



UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

MÁSTER EN INGENIERÍA WEB

Proyecto Fin de Máster

Aplicación web en .NET: Red Social

Autor

Raquel Díaz González

Tutor

Luis Fernando de Mingo López

1 de julio de 2016



AGRADECIMIENTOS

Sin duda, he de decir que ha sido un año duro y de mucho trabajo, pero he pasado unos meses increíbles en los que me he tenido que enfrentar a múltiples situaciones en las que no me había visto envuelta nunca, pero aun así, he sido capaz de compaginar mi trabajo a jornada completa con el máster, y claramente me llevo una gran experiencia.

Quiero expresar mi reconocimiento y agradecimiento a todas aquellas personas que se han interesado por mi progreso y han contribuido a la realización de este Proyecto de Fin de Máster.

A mi tutor del proyecto, Luis, por su apoyo y dedicación.

A mis profesores del máster, que gracias a ellos he aprendido bastantes cosas y me han hecho sentir como en casa.

A mis amigos y compañeros de trabajo, por proporcionarme su interés, su apoyo y su confianza durante todos estos meses.

A mi pareja, por compartir esta nueva etapa junto a mí, por apoyarme en cada momento y por ayudarme en situaciones tanto personales como profesionales.

Y una especial mención a mi familia, por estar siempre a mi lado, tanto en los buenos como en los malos momentos, por apoyarme siempre en mis estudios, y porque sin ellos, claramente esto no habría sido posible.



RESUMEN

Podemos definir las redes sociales on-line como estructuras sociales compuestas por un grupo de personas que comparten un interés común, relación o actividad a través de Internet, donde tienen lugar los encuentros sociales y se muestran las preferencias de consumo de información mediante la comunicación en tiempo real.

El origen de las redes sociales es bastante reciente, se puede decir que surgen en 1995 a manos del estadounidense Randy Conrads con la creación de *classmates.com*. Esta red social buscaba reunir ex compañeros de colegio, o universidades. Y al ver que el proyecto era exitoso, comenzaron a aparecer nuevas redes que pretendían reunir amigos.

Tras el surgimiento de muchas redes sociales, se puede decir que los criterios básicos que debe cumplir una red social son: ser una red de contactos, tener un perfil y ofrecer funcionalidades sociales para interactuar con contenidos. Y de esta forma, se pueden clasificar las redes sociales en 3 tipos: genéricas (*Facebook*), profesionales (*LinkedIn*) y temáticas (*Flickr*).

Según las estadísticas, un 81% de los internautas de 16-55 años utilizan redes sociales, lo que representa más de 15 millones usuarios en nuestro país.

En la siguiente gráfica podemos ver el número de usuarios que utilizan cada red social en el ultimo año.

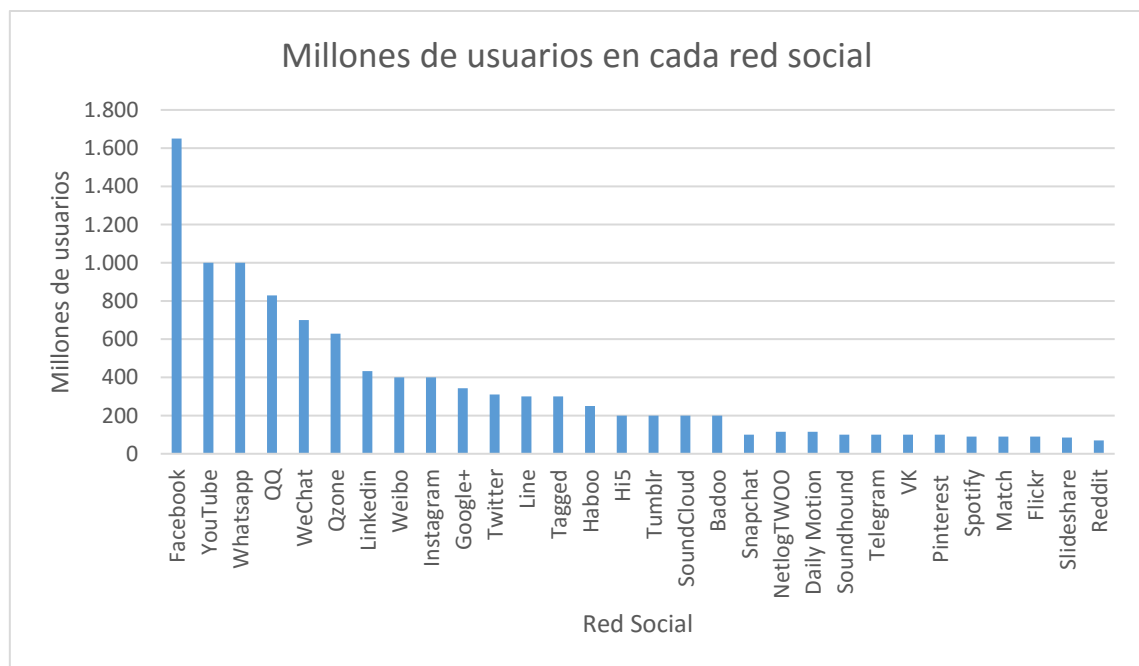


Ilustración 1: Gráfico usuarios / red social

Y debido a este gran crecimiento y uso de las redes sociales, surge la idea de este proyecto.

PALABRAS CLAVE

Desarrollo, cliente, servidor, .NET

ABSTRACT

We can define online social networks as social structures composed of a group of people who share a common interest, relationship or activity over the Internet, where social gatherings take place and consumption of information through communication preferences are displayed in real time.

The origin of social networks is fairly recent, it can be said to arise in 1995 at the hands of Randy Conrads American with the creation of *classmates.com*. This social network sought to meet former teammates from college or universities. And to see that the project was successful, new networks began to appear and they seeking to gather friends.

After the emergence of many social networks, we can said that the basic criteria that social network is: be a network of contacts, a profile and offer social features to interact with content. And in this way, social networks can be classified into 3 types: generic (*Facebook*), professional (*LinkedIn*) and thematic (*Flickr*).

According to statistics, 81% of Internet users of 16-55 years are using social networks, representing more than 15 million users in our country.

In the following chart you can see the number of users who use each social network in the past year.

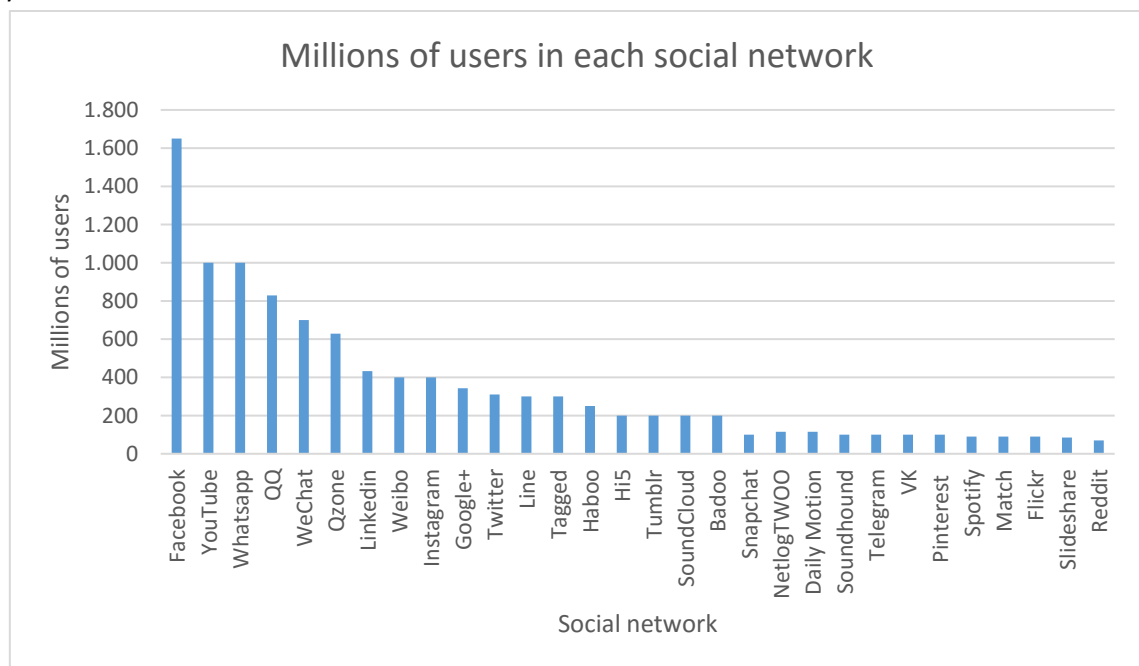


Ilustración 2: Chart users / social network

And because of this great growth and use of social networks, the idea of thisproject.

KEYWORDS

Development, front-end, back-end, .NET



TABLA DE CONTENIDOS

Objetivos.....	15
Tecnologías	17
Entorno de Desarrollo	17
Lenguajes de Programación.....	18
.NET.....	18
ASP.NET.....	18
ASP.NET MVC.....	19
Razor	21
Entity Framework	23
HTML5.....	24
CSS3	24
JavaScript	24
jQuery	24
Bootstrap	25
Base de Datos	26
Metodología	27
Control de Versiones	31
Requisitos	37
Requisitos No Funcionales.....	37
Requisitos Funcionales	38
Análisis	39
Modelo de datos conceptual. Diagrama entidad-relación	39
Diagrama de casos de uso	40
Diseño	41
Modelo de datos lógico. Diagrama relacional	41
Diagrama de clases	43
Prototipado de la Interfaz de Usuario	44
Implementación.....	49
Estructura del proyecto	49
Proceso de desarrollo	52
Evolución de la interfaz de usuario.....	61

Pruebas	93
Conclusiones	95
Posibles Ampliaciones y Mejoras	97
Bibliografía	99



TABLA DE ILUSTRACIONES

Ilustración 1: Gráfico usuarios / red social	5
Ilustración 2: Chart users / social network	7
Ilustración 3: Ejemplo modelo	19
Ilustración 4: Ejemplo vista	19
Ilustración 5: Ejemplo controlador	20
Ilustración 6: Flujo de trabajo ramificado (Git)	32
Ilustración 7: Repositorio GitHub	33
Ilustración 8: Diagrama entidad-relación	39
Ilustración 9: Diagrama casos de uso	40
Ilustración 10: Diagrama relacional	41
Ilustración 11: Diagrama clases (controladores)	43
Ilustración 12: Prototipo inicio aplicación	44
Ilustración 13: Prototipo login	44
Ilustración 14: Prototipo registro	45
Ilustración 15: Prototipo layout	45
Ilustración 16: Prototipo inicio	46
Ilustración 17: Prototipo biografía	46
Ilustración 18: Prototipo perfil	47
Ilustración 19: Prototipo amigos	47
Ilustración 20: Prototipo mensajes	48
Ilustración 21: Creación del proyecto	49
Ilustración 22: Estructura proyecto	49
Ilustración 23: Calendario sprints	52
Ilustración 24: Backlog sprint 1	53
Ilustración 25: Burn down chart sprint 1	53
Ilustración 26: Backlog sprint 2	54
Ilustración 27: Burn down chart sprint 2	54
Ilustración 28: Backlog sprint 3	55
Ilustración 29: Burn down chart sprint 3	56
Ilustración 30: Backlog sprint 4	57
Ilustración 31: Burn down chart sprint 4	57
Ilustración 32: Backlog sprint 5	58
Ilustración 33: Burn down chart sprint 5	58
Ilustración 34: Backlog sprint 6	59
Ilustración 35: Burn down chart sprint 6	59
Ilustración 36: Gráfico tiempo de desarrollo	60
Ilustración 37: Grafo usuarios	61
Ilustración 38: Tabla iconos	61
Ilustración 39: Interfaz inicio aplicación	62
Ilustración 40: Interfaz iniciar sesión	62
Ilustración 41: Interfaz registro	63
Ilustración 42: Interfaz layout	63
Ilustración 43: Interfaz inicio	64

Ilustración 44: Interfaz publicar post.....	64
Ilustración 45: Interfaz publicar comentario	65
Ilustración 46: Interfaz publicar imagen en post	66
Ilustración 47: Interfaz post con imagen	66
Ilustración 48: Interfaz publicar vídeo en post	67
Ilustración 49: Interfaz post con vídeo	67
Ilustración 50: Interfaz con varias acciones.....	68
Ilustración 51: Interfaz biografía.....	69
Ilustración 52: Interfaz publicar post en biografía.....	69
Ilustración 53: Interfaz publicar comentario en un post de la biografía	70
Ilustración 54: Interfaz publicar imagen en un comentario de la biografía	70
Ilustración 55: Interfaz publicar vídeo en un post de la biografía	71
Ilustración 56: Interfaz con varias acciones en biografía.....	72
Ilustración 57: Interfaz perfil por defecto.....	73
Ilustración 58: Interfaz perfil	73
Ilustración 59: Interfaz amigos	74
Ilustración 60: Interfaz búsqueda de amigos	74
Ilustración 61: Interfaz mensajes.....	75
Ilustración 62: Interfaz inicio - administración	75
Ilustración 63: Interfaz ver todos los roles - administración	76
Ilustración 64: Interfaz crear rol - administración	76
Ilustración 65: Interfaz editar rol - administración.....	77
Ilustración 66: Interfaz eliminar rol - administración	77
Ilustración 67: Interfaz ver todos sexos - administración.....	78
Ilustración 68: Interfaz crear sexo - administración	78
Ilustración 69: Interfaz editar sexo - administración.....	79
Ilustración 70: Interfaz ver todos usuarios - administración	80
Ilustración 71: Interfaz crear usuario - administración	80
Ilustración 72: Interfaz editar usuario - administración	81
Ilustración 73: Interfaz eliminar usuario - administración	81
Ilustración 74: Interfaz ver todos posts - administración	82
Ilustración 75: Interfaz crear post - administración	82
Ilustración 76: Interfaz editar post - administración	83
Ilustración 77: Interfaz borrar post - administración	83
Ilustración 78: Interfaz ver todos comentarios - administración	84
Ilustración 79: Interfaz crear comentario - administración.....	84
Ilustración 80: Interfaz editar comentario - administración	85
Ilustración 81: Interfaz eliminar comentario - administración.....	85
Ilustración 82: Interfaz ver todos me gusta - administración.....	86
Ilustración 83: Interfaz crear me gusta - administración.....	86
Ilustración 84: Interfaz editar me gusta - administración	87
Ilustración 85: Interfaz eliminar me gusta - administración.....	87
Ilustración 86: Interfaz ver todos recursos - administración.....	88
Ilustración 87: Interfaz crear recurso - administración	88
Ilustración 88: Interfaz editar recurso (imagen) - administración.....	89



Ilustración 89: Interfaz editar recurso (vídeo) - administración.....	89
Ilustración 90: Interfaz eliminar recurso - administración	89
Ilustración 91: Interfaz laptop L.....	90
Ilustración 92: Interfaz laptop	90
Ilustración 93: Interfaz Tablet.....	90
Ilustración 94: Interfaces mobile L, M, S	91
Ilustración 95: Interfaces menú mobile.....	91

OBJETIVOS

La realización de la asignatura “*Desarrollo de Aplicaciones Web con Tecnologías Propietarias*” desarrolló en mí un gran interés por el nuevo lenguaje de programación que acababa de conocer. Fue entonces cuando tuve claro que mi proyecto debía realizarlo en .NET para poder así desarrollar una aplicación web profundizando un poco más en el lenguaje que acababa de conocer.

El objetivo principal con el que se define este Proyecto Fin de Máster es desarrollar una red social que permita realizar las funciones básicas que debe tener toda red social.

Otro factor importante es que mi dedicación a jornada completa en una empresa hacía que el desarrollo del proyecto por las tardes fuese prácticamente imposible, por lo que tenía un nuevo reto que cumplir. Necesitaba una buena organización siguiendo una metodología de trabajo adecuada para poder terminar el proyecto a tiempo teniendo en cuenta que prácticamente solo disponía de los fines de semana.

Durante todas las asignaturas del máster nos han insistido bastante en definir una buena arquitectura, mantener un buen código, seguir unas reglas de estilo... y todo eso estaba en mi lista de objetivos.

TECNOLOGÍAS

En este apartado comentaremos el entorno de desarrollo, los lenguajes de programación y la base de datos.

ENTORNO DE DESARROLLO

Microsoft Visual Studio es un entorno de desarrollo integrado (*IDE*) para sistemas operativos *Windows*, extensible y gratuito para crear aplicaciones modernas para *Windows*, *Android* e *iOS*, además de aplicaciones web y servicios en la nube.

Visual Studio Community es gratuito para desarrolladores individuales, proyectos de código abierto, investigación académica, educación y pequeños equipos profesionales



Entre sus **características** destacan:

- Flexibilidad
- Productividad
- Extensiones
- Diferentes lenguajes (*C#, Visual Basic, F#, C++, HTML, JavaScript, Python...*)
- Herramientas web (*ASP.NET, Node.js, Python, JavaScript, AngularJS, jQuery, Bootstrap, Django y Backbone.js*)
- Herramientas de codificación eficaces
- Depuración avanzada
- Aplicaciones para dispositivos
- Integración con *Git*
- Beneficios (herramientas de desarrollo gratuitas, aprendizaje de *Pluralsight*, créditos de *Azure*)
- Uso (para usuarios individuales, para organizaciones y para organizaciones que no sean empresariales)

LENGUAJES DE PROGRAMACIÓN

.NET

Microsoft .NET es el conjunto de nuevas tecnologías en las que *Microsoft* ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el



software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

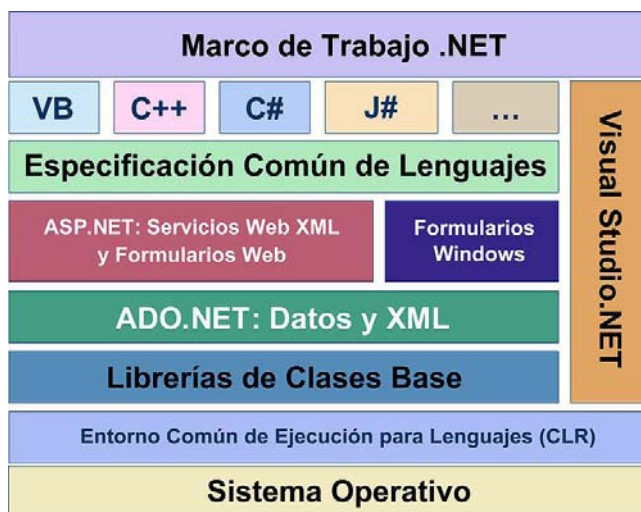
.NET podría considerarse una respuesta de *Microsoft* al creciente mercado de los negocios en entornos Web, como competencia a la plataforma *Java* de *Oracle Corporation* y a los diversos *frameworks* de desarrollo web basados en PHP, ya que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware para permitir un rápido desarrollo de aplicaciones.

ASP.NET



ASP.NET apareció en enero de 2002 y es una plataforma web que proporciona todos los servicios necesarios para compilar aplicaciones web empresariales basadas en servidor. *ASP.NET* está compilado en *.NET Framework*, por lo que todas las características de *.NET Framework* están disponibles en las aplicaciones ASP.NET y las aplicaciones se pueden escribir en cualquier lenguaje que sea compatible con *Common Language Runtime (CLR)*.

En la siguiente imagen se puede ver el conjunto que engloba .NET



ASP.NET MVC

ASP.NET MVC es un *framework* de aplicaciones web que implementa el patrón modelo-vista-controlador (MVC) y proporciona una alternativa al modelo de formularios *Web Forms* de *ASP.NET* para crear aplicaciones web, en el que se incluyen los siguientes componentes:

- **Modelos.** Los objetos de modelo son las partes de la aplicación que implementan la lógica del dominio de datos de la aplicación. A menudo, los objetos de modelo recuperan y almacenan el estado del modelo en una base de datos.

```
public partial class Post
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    public Post()
    {
        this.Comments = new HashSet<Comment>();
        this.Likes = new HashSet<Like>();
        this.Medias = new HashSet<Media>();
    }

    public int Id { get; set; }
    [Required]
    [Display(Name = "Texto")]
    public string Text { get; set; }
    [Required]
    [Display(Name = "Fecha y hora")]
    public System.DateTime DateTime { get; set; }
    [Required]
    [Display(Name = "Usuario")]
    public string UserId { get; set; }

    public virtual AspNetUser AspNetUser { get; set; }
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Comment> Comments { get; set; }
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Like> Likes { get; set; }
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Media> Medias { get; set; }
}
```

Ilustración 3: Ejemplo modelo

- **Vistas.** Las vistas son los componentes que muestra la interfaz de usuario de la aplicación. Normalmente, esta interfaz de usuario se crea a partir de los datos de modelo.

```
@using RedSocial.Models
@using Microsoft.AspNet.Identity
@model RedSocial.AspNetUser

<div class="cover profile">
    <div class="wrapper">
        <div class="image">
            
        </div>
    </div>
    <div class="cover-info">
        <div class="avatar">
            
        </div>
        <ul class="cover-nav">
            <li class="active"><a> @Html.DisplayFor(model => model.UserName)</a></li>
            <li><a><i class="fa fa-fw icon-ship-wheel"></i> @Html.DisplayFor(model => model.Posts.Count) Posts</a></li>
            <li><a><i class="fa fa-fw fa-heart"></i> @Html.DisplayFor(model => model.Likes.Count) Me gusta</a></li>
            <li><a><i class="fa fa-fw fa-comments"></i> @Html.DisplayFor(model => model.Comments.Count) Comentarios</a></li>
            <li><a><i class="fa fa-fw fa-users"></i> @Html.DisplayFor(model => model.AspNetUsers.Count) Amigos</a></li>
        </ul>
    </div>
</div>
```

Ilustración 4: Ejemplo vista

- **Controladores.** Los controladores son los componentes que controlan la interacción del usuario, trabajan con el modelo y por último seleccionan una vista para representar la interfaz de usuario.

```
// GET: Roles/Create
[Authorize(Roles = GlobalConstants.RoleAdmin)]
public ActionResult Create()
{
    return View();
}
```

Ilustración 5: Ejemplo controlador

El modelo MVC le ayuda a crear aplicaciones que separan los diferentes aspectos de la aplicación (lógica de entrada, lógica de negocios y lógica de la interfaz de usuario), a la vez que proporciona un vago acoplamiento entre estos elementos.

Ventajas de una aplicación web basada en MVC

- Facilita la administración de la complejidad, al dividir una aplicación en el modelo, la vista y el controlador.
- No usa el estado de vista ni formularios basados en servidor. Esto hace que el marco de MVC sea ideal para los desarrolladores que deseen un control completo sobre el comportamiento de una aplicación.
- Usa un modelo de controlador frontal que procesa las solicitudes de la aplicación web a través de un controlador único, lo que permite diseñar una aplicación que admite una infraestructura de enrutamiento avanzada.
- Proporciona una mayor compatibilidad con el desarrollo basado en pruebas (TDD).

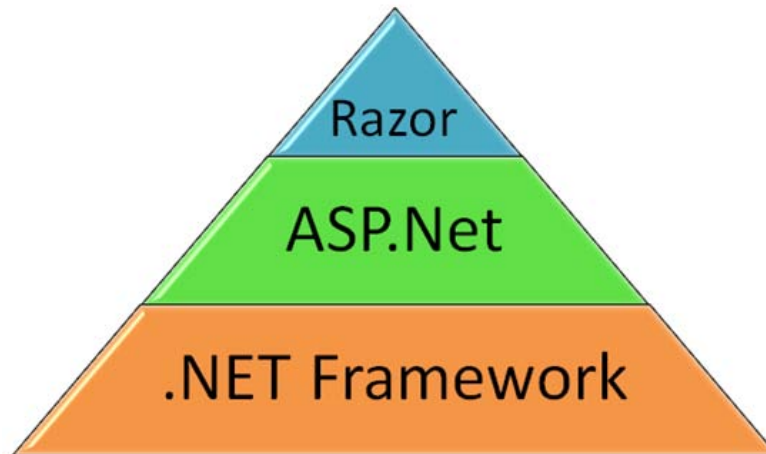
Características del marco de ASP.NET MVC

- Separación de tarea, facilidad para pruebas y desarrollo basado en pruebas (TDD).
- Un marco extensible y conectable. Los componentes están diseñados para que se puedan reemplazar o personalizar con facilidad.
- Amplia compatibilidad para el enrutamiento de *ASP.NET*. Las direcciones URL no tienen que incluir las extensiones de los nombres de archivo y están diseñadas para admitir patrones de nombres de direcciones URL que funcionan bien para la optimización del motor de búsqueda (SEO) y el direccionamiento de transferencia de estado representacional (REST).
- Compatibilidad para usar el marcado en archivos de *ASP.NET* existente (.aspx), de controles de usuario (.ascx) y de páginas maestras (.master) como plantillas de vista.
- Compatibilidad con las características de *ASP.NET* existentes (autenticación de formularios, autenticación de Windows, autorización para URL, pertenencia y los roles, caching, administración de estados de sesión y perfil, seguimiento de estado, sistema de configuración y la arquitectura de proveedor).

RAZOR

Razor proporciona una sintaxis de programación simple para escribir código en páginas web donde el código basado en servidor se incrusta en el formato *HTML* de las páginas web. El código de *Razor* se ejecuta en el servidor antes de que la página se envíe al explorador.

Las páginas web que incluyen contenido de *Razor* tienen una extensión de archivo especial (.cshtml o .vbhtml). El servidor reconoce estas extensiones, ejecuta el código que está marcado con sintaxis de *Razor* y, a continuación, envía la página resultante al explorador.



Con la sintaxis que proporciona *Razor* se puede hacer, entre otras cosas, lo siguiente:

- Construir un link

```
<p>  
    @Html.ActionLink("Nuevo", "Create", null, new { @class = "btn btn-primary" })  
</p>
```

- Hacer formularios

```
@using (Html.BeginForm())  
{  
    @Html.AntiForgeryToken()  
  
    <div class="form-horizontal">  
  
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })  
        <div class="form-group">  
            @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })  
            <div class="col-md-10">  
                @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })  
                @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })  
            </div>  
        </div>  
  
        <div class="form-group">  
            <div class="col-md-offset-2 col-md-10">  
                <button type="submit" class="btn btn-primary"><i class="fa fa-send-o"></i> Guardar</button>  
            </div>  
        </div>  
    </div>  
}
```

- Incluir archivos parciales

```
<div class="col-xs-12 col-md-6 col-lg-6 col-lg-offset-3 col-md-offset-3">
    @Html.Action("CreatePostPartial")
</div>
```

- Usar variables

```
<div class="collapse" id="collapseCommentVideo_@ViewBag.PostId" style="padding-top:10px;">
    <input type="text" name="video" class="form-control text-box single-line" placeholder="URL del vídeo" />
</div>
```

- Usar objetos

```
<footer class="footer">
    <strong>Proyecto Fin de Máster</strong> - Raquel Díaz González - &copy; @DateTime.Now.Year
</footer>
```

- Uso de bucles

```
@if (User.IsInRole(GlobalConstants.RoleAdmin))
{
    <li>
        <a href="@Url.Action("Index", "Admin")">
            Ir a administración...
        </a>
    </li>
}
```

Existen bastantes formas para pasar datos del controlador a la vista (*ViewData*, *ViewBag*, *PartialView*, *TempData*, *ViewModel*, *Tuple*...) pero en este proyecto solo se han usado 3:

1. El controlador redirige a una vista sin pasar datos

```
return View();
```

2. El controlador redirige a una vista pasando una variable

```
return View(db.AspNetRoles.ToList());
```

```
@model IEnumerable<RedSocial.AspNetRole>
```

```
@foreach (var item in Model)
{
```

3. El controlador redirige a una vista usando ViewBag

```
ViewBag.Posts = db.Posts.ToList();
return View();
```

```
<h3>@ViewBag.Posts.Count</h3>
```

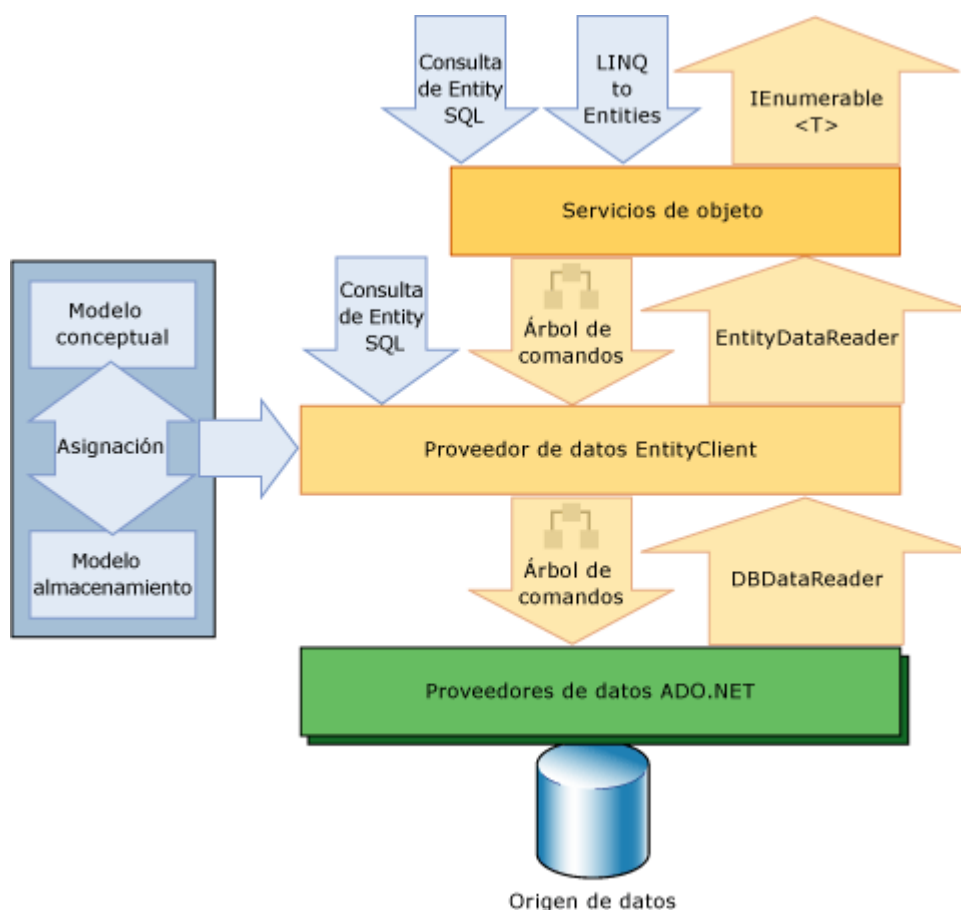

ENTITY FRAMEWORK

Entity Framework es un conjunto de tecnologías de *ADO.NET* que permiten el desarrollo de aplicaciones de software orientadas a datos. Permite trabajar con datos en forma de objetos y propiedades específicos del dominio, sin tener que preocuparse por las tablas y columnas de la base de datos subyacente donde se almacenan estos datos. Con *Entity Framework*, los desarrolladores pueden trabajar en un nivel mayor de abstracción cuando tratan con datos, y pueden crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales.



Entity Framework incluye el proveedor de datos de *EntityClient*, el cual, administra las conexiones, traduce las consultas de entidad en consultas específicas del origen de datos y devuelve un lector de datos que *Entity Framework* usa para materializar los datos de la entidad en los objetos.

El diagrama siguiente muestra la arquitectura de *Entity Framework* para el acceso a datos:



[https://msdn.microsoft.com/es-es/library/bb399567\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/bb399567(v=vs.110).aspx)

HTML5

Es el lenguaje de marcado de hipertexto para los sitios web desarrollados por el *World Wide Web Consortium (W3C)*.



CSS3

Es un lenguaje utilizado para definir y crear la presentación de un documento estructurado escrito en *HTML* o *XML*. Su idea se basa en separar la estructura del documento de la presentación.



JAVASCRIPT

Es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, orientado a objetos y con tipado dinámico que permite mejoras en la interfaz de usuario.



JQUERY

jQuery es una librería de *JavaScript* que permite simplificar la manera de interactuar con los documentos *HTML*, manipular el árbol *DOM*, manejar eventos, desarrollar animaciones y agregar interacción con la técnica *AJAX* a páginas web



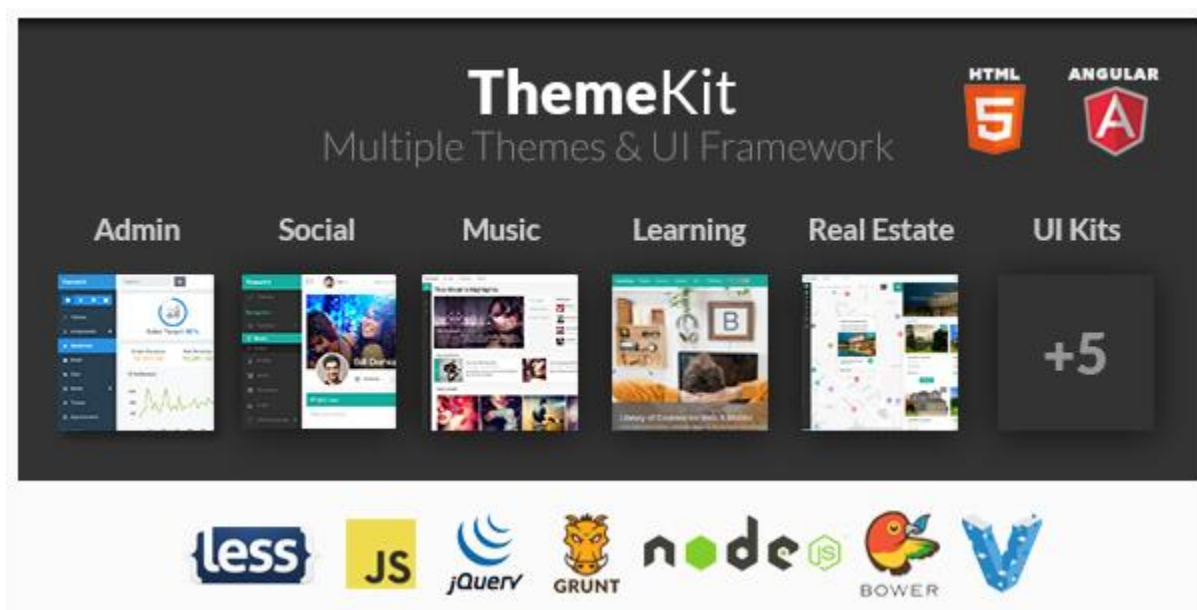
BOOTSTRAP

Bootstrap es un *framework* para el diseño de sitios y aplicaciones web. Sus elementos están basados en *HTML*, *CSS* y extensiones de *JavaScript*. Entre sus características más importantes se pueden destacar las siguientes:

- Incluye soporte para la mayor parte de navegadores.
- Facilita el trabajo gracias a los sistemas de rejilla (*grids*)
- Diseño adaptable
- Código abierto y con una gran comunidad.



Para facilitar el desarrollo de la interfaz de usuario se ha utilizado la siguiente plantilla de *Bootstrap*:



<http://themeforest.net/item/themekit-bootstrap-admin-theme-kit/6767145>

BASE DE DATOS

La aplicación desarrollada requiere el almacenamiento de una gran cantidad de información que será utilizada por los diferentes usuarios.

Es por ello que surge la necesidad de crear una base de datos en la que se almacene toda la información. Hay multitud de bases de datos, pero en este caso se dispone de una base de datos *Microsoft SQL Server*.



Microsoft SQL Server es un sistema de gestión de bases de datos relacionales cuyo servicio principal es almacenar, procesar y proteger los datos, que proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones consumidoras de datos más exigentes de su empresa.

Para realizar operaciones sobre la base de datos desde los controladores se utiliza *Entity Framework* porque tenemos acceso a las tablas y sus atributos simplemente instanciando el objeto.

```
private Entities db = new Entities();
```

Por tanto, para hacer una consulta, podremos hacer algo como esto:

```
Post post = db.Posts.Find(id);
```

También existe la posibilidad de usar expresiones *lamda*:

```
var posts_ordered = aspNetUser.Posts.OrderByDescending(x => x.DateTime).ToList();
```

O realizar consultas con *LINQ*:

```
var results = from c in aspNetUsers
               where c.UserName.Contains(userName)
               select c;
```

LINQ (Language-Integrated Query) es un conjunto de características que agrega capacidades de consulta eficaces.

METODOLOGÍA

El **proceso de desarrollo software** es el conjunto total de actividades necesarias para transformar los requerimientos del cliente en un conjunto consistente de artefactos que representan un producto software y, en un momento posterior, para transformar los cambios de estos requerimientos en una nueva versión del producto software.

Se define **metodología** como aquella disciplina que indica qué métodos y técnicas hay que usar en cada fase del ciclo de vida de desarrollo del proyecto. A lo largo de la historia se han utilizado las metodologías convencionales hasta que surgieron las metodologías ágiles como una alternativa a estas metodologías.

Hoy en día existen bastantes metodologías de desarrollo:

- Metodología Top-Down (diseño, sin requisitos)
- Metodología Bottom-Up (diseño, sin requisitos)
- Metodología en Cascada
- Metodología en V (doble Cascada)
- Metodología Staffing
- Metodologías ágiles (XP, Crystal Clear, Kanban, Scrum, ...)
- Metodologías del Proceso Unificado (UP) de Rationale

Pero para el desarrollo de este proyecto se ha utilizado la metodología ágil de Scrum.

Características de las metodologías ágiles

- Proceso iterativo e incremental. Fases en paralelo
- Flexible. Adaptable. Mejora continua
- Entregas pequeñas y frecuentes de software, que siempre aportan valor
- Adecuado cuando el usuario no conoce al detalle todas las necesidades al principio, y, además, serán cambiantes
- El usuario está involucrado constantemente en el proyecto. Comunicación fluida
- Los requisitos se van desgranando cuando se van a abordar
- Ámbito y costes varían según evoluciona el proyecto
- La documentación no es prioritaria

Scrum es una metodología ágil para la gestión de proyectos, en la que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto, adaptándose a los cambios en los requisitos. Por tanto, se puede decir que los pilares de *Scrum* son básicamente dos, el ciclo de vida iterativo e incremental y las diversas reuniones a lo largo del proyecto.



- **Artefactos**

- *Sprint (iteración)*: periodo de tiempo durante el que se desarrolla un incremento de funcionalidad. Durante el sprint no se puede modificar el trabajo que se ha acordado en el Backlog.
- *Historias de usuario*: descripciones de las funcionalidades que va a tener el software. Serán el resultado de la colaboración entre el cliente y el equipo e irán evolucionando durante toda la vida del proyecto.
- *Product backlog*: listado con los requisitos del sistema que se mantiene durante todo el ciclo de vida
- *Sprint backlog*: tareas determinadas por el equipo para realizar en un sprint y lograr al final del mismo un incremento de la funcionalidad.
- *Burn Down Chart*: es un diagrama en el que se representan en el eje de ordenadas el número total de requisitos o tareas pendientes y en el eje de abscisas el tiempo.
- *Release*: versión que se pasa a producción.
- *Roadmap*: mapa de la *release*.

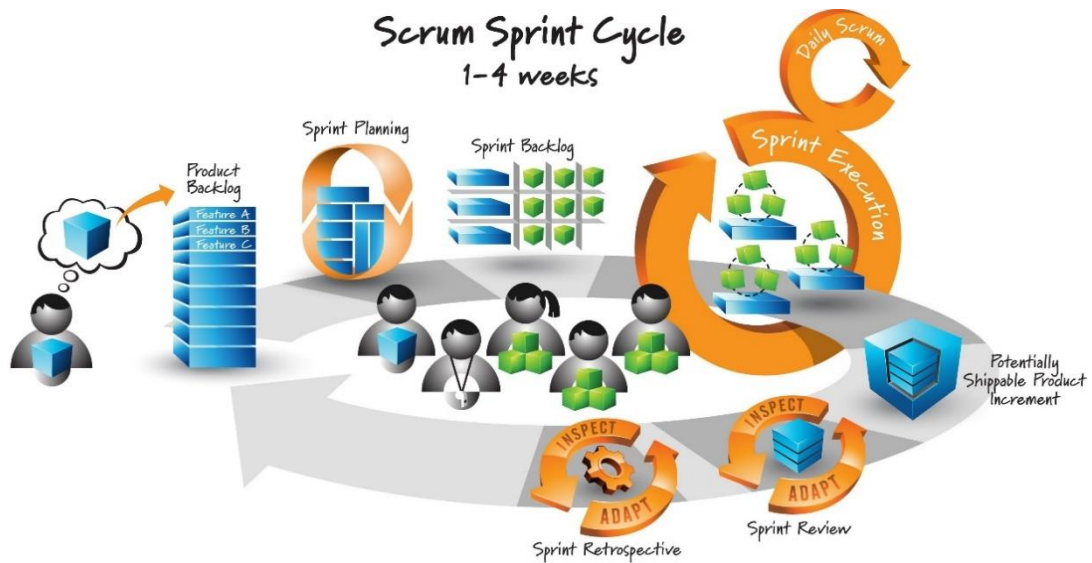
- **Roles**

- *Product owner*: tiene una visión del producto, las entregas (releases) y el sprint. Sus funciones son mantener y priorizar las funcionalidades, el product backlog, las historias de usuario, respetar las estimaciones del equipo y aceptar o rechazar los resultados del trabajo del equipo
- *Scrum Master*: es el responsable de promover los valores y prácticas de Scrum, se asegura de que el equipo es completamente funcional y productivo y no forma parte del equipo de desarrollo
- *Team*: formado por cualquier rol que sea necesario para realizar el incremento del producto (desarrolladores, analistas, diseñadores, testers, equipo de QA, y el product owner. Sus funciones son cumplir con el sprint, realizar estimaciones y planificar su trabajo

- **Reuniones**

- *Sprint planning (planificación del sprint)*: se realiza la primera mañana de cada sprint con la participación de todos los miembros del equipo de desarrollo y el product owner. En ella, se analizan las historias de usuario para identificar las tareas, los criterios de aceptación, la estimación...
- *Daily meeting (reunión diaria)*: se realiza cada día a la misma hora y en el mismo lugar y en ella, cada miembro del equipo responde a las preguntas de ¿qué hiciste ayer?, ¿qué harás hoy? y ¿Hay algún problema para la realización de tus tareas?
- *Sprint review (reunión de revisión)*: al finalizar el sprint, el equipo presenta lo que se ha realizado (normalmente con una demo)
- *Sprint retrospective (retrospectiva del sprint)*: en esta reunión participa todo el equipo para analizar cómo se ha trabajado y se identifican qué se ha hecho bien, qué se ha hecho mal y qué puede mejorarse.

En la siguiente imagen se puede ver un resumen gráfico con todos los elementos explicados anteriormente.



<http://springtimesoft.co.nz/agile-software-development-scrum/>

CONTROL DE VERSIONES

Un **sistema de control de versiones** se utiliza para el almacenamiento y gestión de los diversos cambios que se realizan sobre el código fuente, pudiendo recuperar cualquier versión en cualquier momento. Existen diferentes tipos:

- Distribuidos. Aumenta la flexibilidad pero complica la sincronización y gestión (*Git*, *Mercurial*...)
- Centralizados. Dependiente de un responsable. Facilita la gestión pero reduce la potencia y flexibilidad (*CVS*, *Subversion*...)

Para el desarrollo de este proyecto se ha utilizado *Git* como sistema de control de versiones.

Git nació en 2005 cuando la comunidad que desarrollaba el núcleo de Linux comenzó a desarrollar su propia herramienta para gestionar los cambios en el software ya que la compañía que desarrollaba *BitKeeper* (DVCS propietario) se vino abajo.



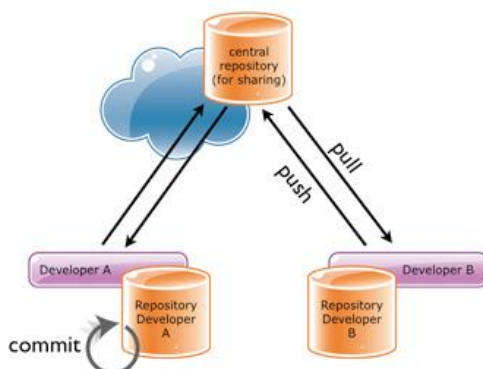
GitHub

En 2008 nació **GitHub** como una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones *Git* y hoy cuenta con más de 9 millones de usuarios registrados, más de 200 millones de visitas al mes y ha sido valorado en más de 2000 millones de dólares

Entre las características más relevantes de *Git*, podemos destacar las siguientes:

- Fuerte apoyo al desarrollo no lineal
- Gestión distribuida.
- Los almacenes de información pueden publicarse por *HTTP*, *FTP*, *rsync* o mediante un protocolo nativo
- Gestión eficiente de proyectos grandes
- Muy fiable

En la siguiente imagen se puede ver el funcionamiento de *Git*.



El proyecto se publica en un repositorio remoto (*central repository*) y cada usuario se conecta a él, y tiene su área de trabajo local (*repository developer*) que es una copia para que se pueda modificar.

El **flujo de trabajo** de un sistema de control de versiones indica cómo se relacionan los miembros del equipo para colaborar entre sí en el desarrollo del software colaborativo. Existen diferentes tipos:

- Flujo de trabajo centralizado (*Centralized Workflow*). Todos comparten el código de un repositorio central, o en Git, todos comparten la única rama
- Flujo de trabajo con bifurcación (*Forking Workflow*). Normalmente, se utiliza en proyectos abiertos en los que el usuario bifurca el proyecto realizando una copia completa del repositorio, realiza las ampliaciones y realiza una petición de agregación (*pull request*). Se abre una discusión, y el dueño del repositorio decide la fusión.
- Flujo de trabajo ramificado (*Git Workflow*). Gracias a los repositorios distribuidos, ha tomado auge por la gestión efectiva de las ramas

En este proyecto se ha seguido la estrategia de flujo de trabajo ramificado (*Git Workflow*)

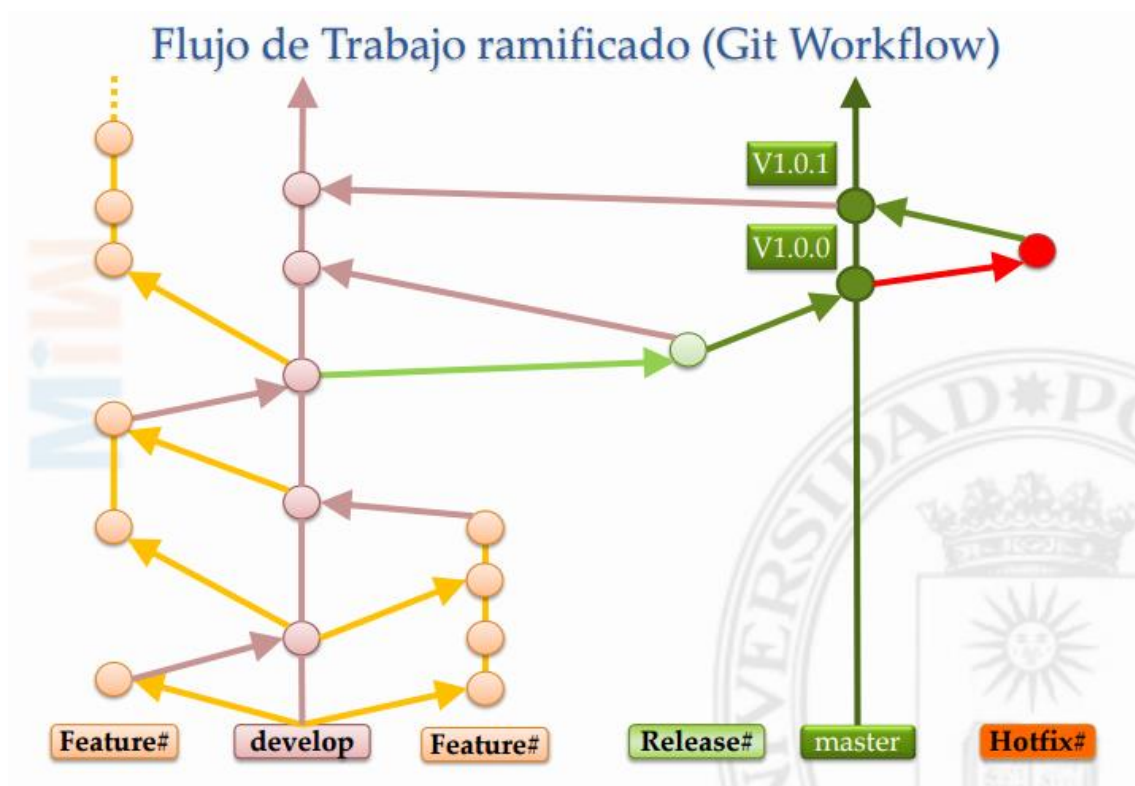
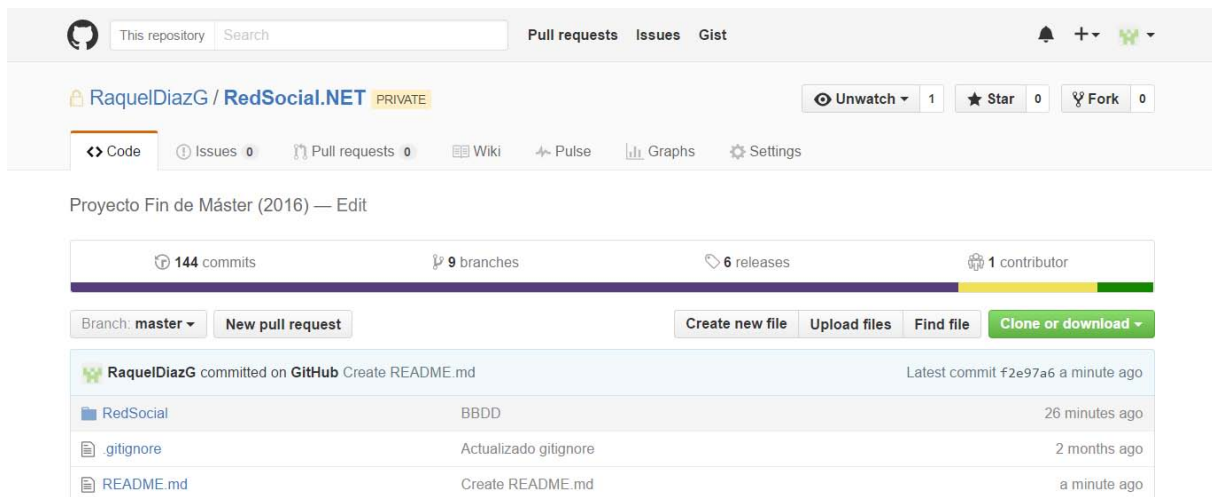


Ilustración 6: Flujo de trabajo ramificado (Git)

- Rama master: apunta al último commit de producción
- Rama develop: es la rama de desarrollo, siempre estable, código que cumple los requisitos de calidad y con todos los test superados
- Ramas feature#: parte de develop, se añaden las nuevas características y vuelve a develop cuando vuelve a ser estable
- Ramas release#: se utiliza para estabilizar un código para salir a producción. Una vez terminada, se fusiona con master y con develop.
- Ramas hotfix#: se utilizan para corregir errores (bugs) en el código en producción

El repositorio de *GitHub* es el que se puede ver en la siguiente imagen:



Y el fichero README.md contiene lo siguiente:

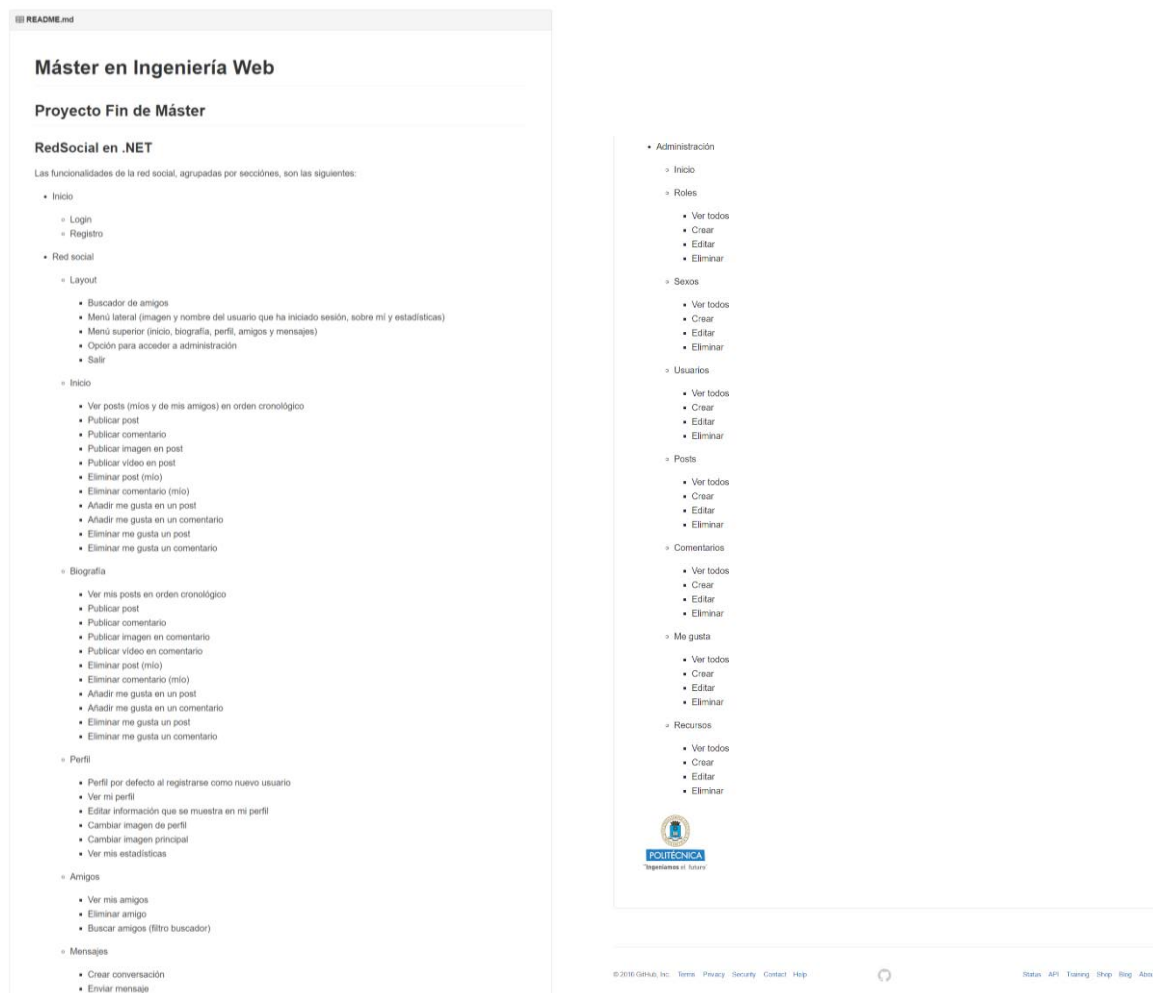
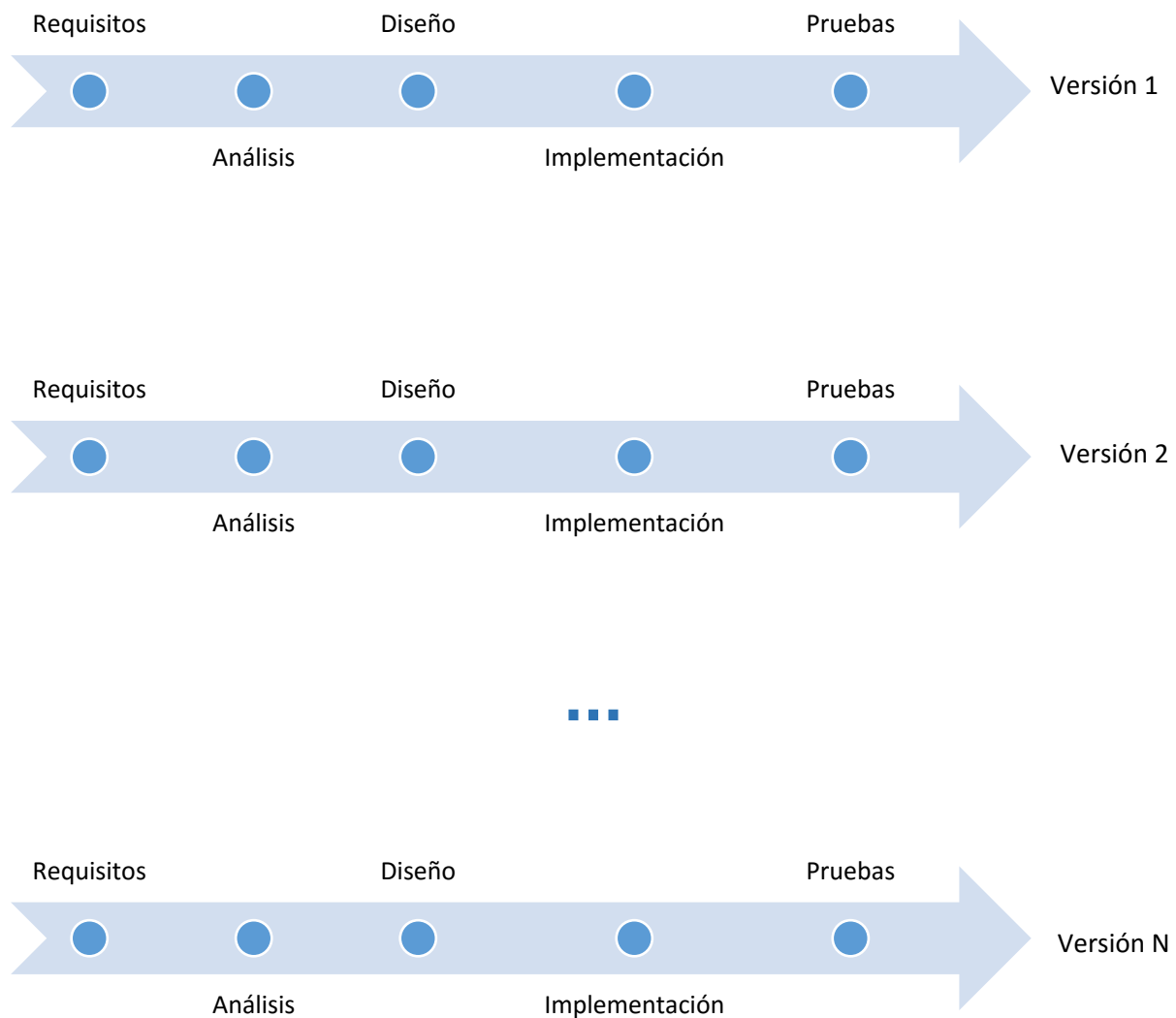


Ilustración 7: Repositorio GitHub

Para explicar los siguientes apartados se realiza un recorrido por las fases del ciclo de vida del software para explicar cómo se ha llevado a cabo cada una de las fases.



REQUISITOS

La **disciplina de requisitos** es el flujo de trabajo, incluyendo actividades, trabajadores y documentos, cuyo propósito principal es dirigir el desarrollo hacia el sistema correcto al describir los requisitos del sistema así que pueda alcanzarse un acuerdo entre los clientes, usuarios y desarrolladores sobre lo que el sistema debería hacer:

- Establecer y mantener el acuerdo entre los clientes y otros interesados sobre lo que el sistema debería hacer
- Proveer una mejor comprensión de los requisitos del sistema
- Definir los límites del sistema
- Proveer las bases para planificar los aspectos técnicos del desarrollo
- Proveer las bases para estimar los costes y tiempos para desarrollar el sistema

REQUISITOS NO FUNCIONALES

Los **requisitos no funcionales** corresponden a las restricciones de los servicios o funciones ofrecidas por el sistema y se aplican al sistema en su totalidad. Surgen de las necesidades del usuario debido a restricciones de presupuesto, políticas de la organización, necesidad de interoperatividad....

Los requisitos no funcionales identificados para esta aplicación son los siguientes:

- Rendimiento: se pretende que la aplicación tenga tiempos de respuesta rápidos, ya que es necesario hacer operaciones que conlleven una carga de tiempo mínima.
- Disponibilidad: se pretende que la aplicación esté siempre disponible garantizando el acceso a la información a los usuarios.
- Seguridad: existe autenticación de usuarios mediante un cifrado de contraseña.
- Operatividad: se pretende que el software produzca los resultados esperados, y de no ser así, deben estar controlados para mostrar información al usuario.
- Escalabilidad: el sistema está preparado para añadir nuevas funcionalidades sin perjudicar a las actuales.

REQUISITOS FUNCIONALES

Los **requisitos funcionales** son los servicios que debe proporcionar la aplicación desarrollada.

A continuación se muestra el listado de requisitos funcionales:

- Login
- Registro
- Red social
 - Añadir/eliminar post
 - Ver mis posts
 - Ver posts de mis amigos
 - Ver mis amigos
 - Añadir/eliminar amigos
 - Buscador de amigos (todos y por nombre)
 - Ver mi perfil
 - Ver el perfil de un usuario
 - Editar información de mi perfil
 - Añadir/eliminar comentario en un post
 - Subida de imágenes (imagen de perfil, imagen principal)
 - Añadir imágenes y vídeos en un post
 - Añadir imágenes y vídeos en un comentario
 - Añadir/eliminar me gusta un post
 - Añadir/eliminar me gusta un comentario
 - Contador de comentarios y me gusta
 - Enviar mensajes privados entre amigos
- Administración
 - Ver/Añadir/Editar/Eliminar sexos
 - Ver/Añadir/Editar/Eliminar roles
 - Ver/Añadir/Editar/Eliminar usuarios
 - Ver/Añadir/Editar/Eliminar posts
 - Ver/Añadir/Editar/Eliminar comentarios
 - Ver/Añadir/Editar/Eliminar me gusta
 - Ver/Añadir/Editar/Eliminar recursos

ANÁLISIS

La **disciplina de análisis** es el flujo de trabajo, incluyendo trabajadores, actividades y documentos, cuyo principal objetivo es analizar los requisitos a través de su refinamiento y estructura para realizar una comprensión más precisa de los requisitos, una descripción de los requisitos que es fácil de mantener y ayuda a estructurar el sistema.

MODELO DE DATOS CONCEPTUAL. DIAGRAMA ENTIDAD-RELACIÓN

El **modelo conceptual** está orientado a la descripción de estructuras de datos y restricciones de integridad.

Un **diagrama entidad-relación** es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema junto con sus interrelaciones y propiedades.

A continuación, se muestra el diagrama E/R realizado con la herramienta *Dia* disponible en <http://www.dia-installer.de/>

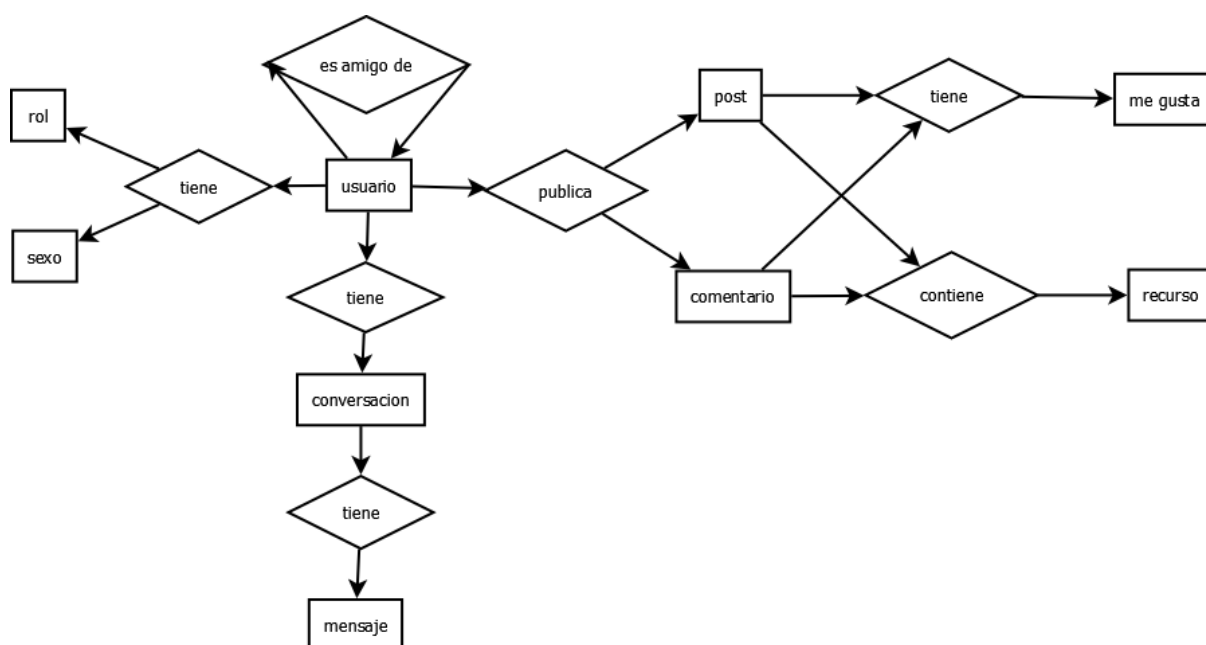


Ilustración 8: Diagrama entidad-relación

Los artefactos cuadrados se convertirán posteriormente en tablas de la base de datos, y los artefactos en forma de rombo serán las relaciones entre las tablas.

DIAGRAMA DE CASOS DE USO

Un **caso de uso** especifica el comportamiento deseado del sistema, es decir, un conjunto de secuencias de acciones, incluyendo variantes, que el sistema puede ejecutar y que produce un resultado observable de valor para un particular actor.

A continuación, se muestra el **diagrama de casos de uso** de la aplicación para describir las funciones que se pueden realizar en el sistema dependiendo del usuario que esté interactuando. Dicho diagrama ha sido realizado con la herramienta *MagicDraw* disponible en <http://www.nomagic.com/products/magicdraw.html>

Para facilitar el entendimiento del diagrama se han repetido los actores, pero en realidad, solo existen tres (admin, usuario logueado y usuario no logueado)

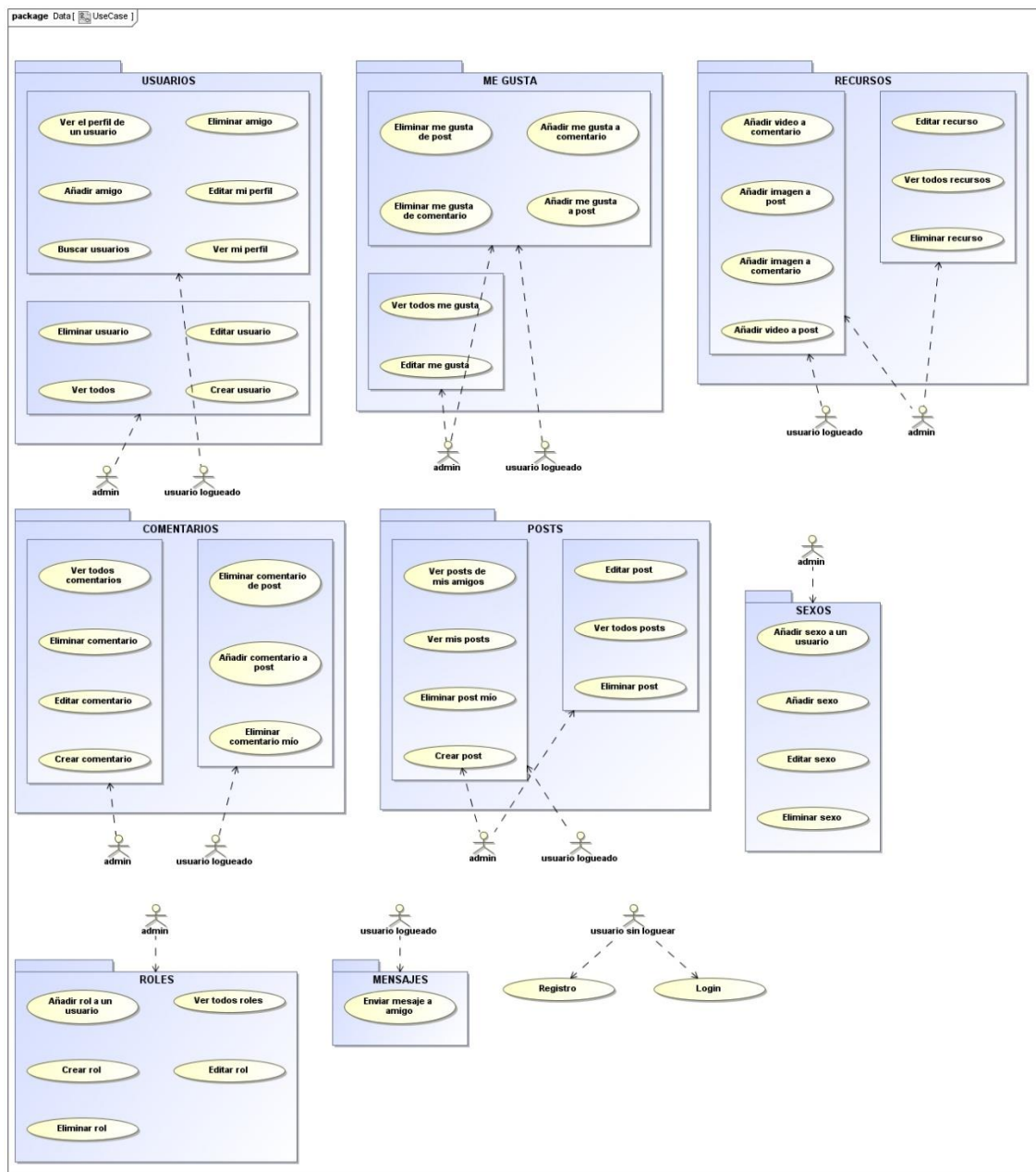


Ilustración 9: Diagrama casos de uso

La **disciplina de diseño** es el flujo de trabajo, incluyendo actividades, trabajadores y documentos, cuyo principal propósito es desarrollar modelos enfocados en los requisitos no funcionales y en el dominio de la solución para preparar para la implementación y pruebas del sistema.

Una vez definido el modelo conceptual, se transforma a un **modelo de datos lógico** en el que se describe la estructura de datos para una base de datos relacional.

La estructura y relación entre las tablas se muestra en el siguiente diagrama:

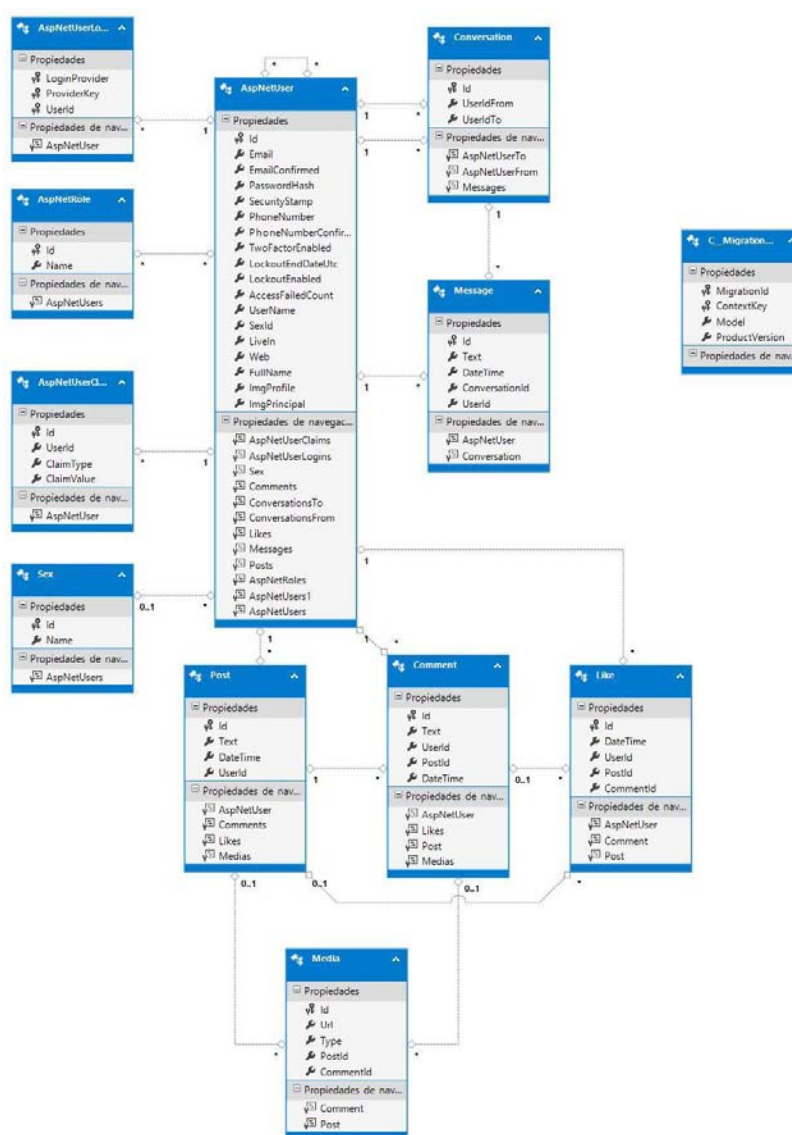


Ilustración 10: Diagrama relacional

Las tablas con el prefijo Asp (*AspNetUser*, *AspNetUserLogin*, *AspNetUserClaim*, *AspNetRole*) y la tabla *C_MigrationHistory* han sido generadas automáticamente cuando se ha creado el proyecto ASPNET MVC.

La tabla principal es la de usuarios (*AspNetUser*) y se relaciona con el resto de tablas.

Un usuario tiene asociado un sexo (*Sex*) y un rol (*AspNetRole*).

Dos usuarios pueden enviarse mensajes, y para ello un usuario (*AspNetUser*) tiene una conversación (*Conversation*) con otro usuario (*AspNetUser*), y a su vez un usuario (*AspNetUser*) envía un mensaje (*Message*) a una conversación (*Conversation*).

Un usuario (*AspNetUser*) tiene asociados todos los posts que ha publicado (*Post*)

Un usuario (*AspNetUser*) puede añadir un comentario (*Comment*) a un post (*Post*)

Un usuario (*AspNetUser*) puede indicar que le gusta un post (*Post*) o un comentario (*Comment*)

Un usuario (*AspNetUser*) puede añadir imágenes o vídeos (*Media*) a un post (*Post*) o a un comentario (*Comment*).

DIAGRAMA DE CLASES

Un **diagrama de clases** es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases.

En la siguiente imagen se pueden ver todos los controladores de la aplicación.

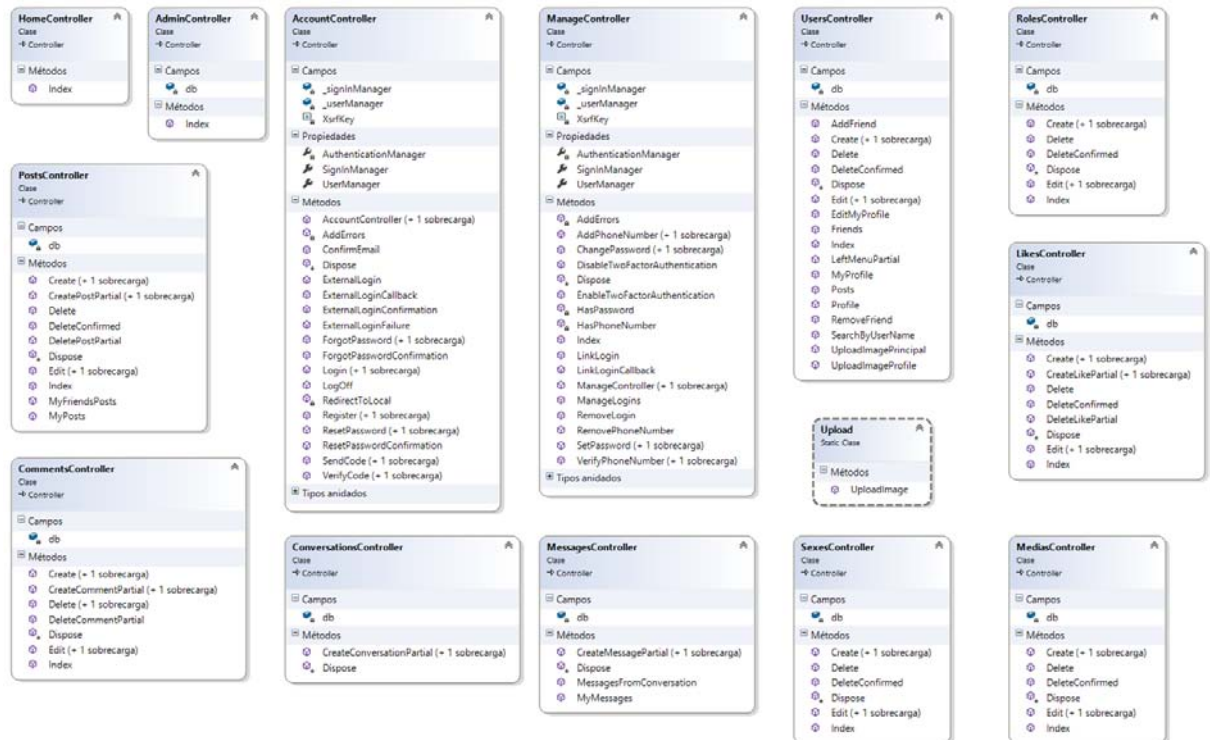


Ilustración 11: Diagrama clases (controladores)

PROTOTIPADO DE LA INTERFAZ DE USUARIO

Una vez analizado el problema y capturados los requisitos funcionales, se realiza una **fase de prototipado**, realizando un primer diseño de la interfaz. Más adelante se indicará cómo se añadieron nuevas funcionalidades haciendo que esta interfaz evolucionase.

El desarrollo de los prototipos se ha realizado herramienta *Balsamiq* disponible en <https://balsamiq.com/>

Al entrar por primera vez en la aplicación se mostrará la interfaz de presentación a través de la cual se podrá acceder tanto al login como al registro.



Ilustración 12: Prototipo inicio aplicación

Un usuario ya registrado, podrá acceder a la aplicación introduciendo su correo electrónico y su contraseña.

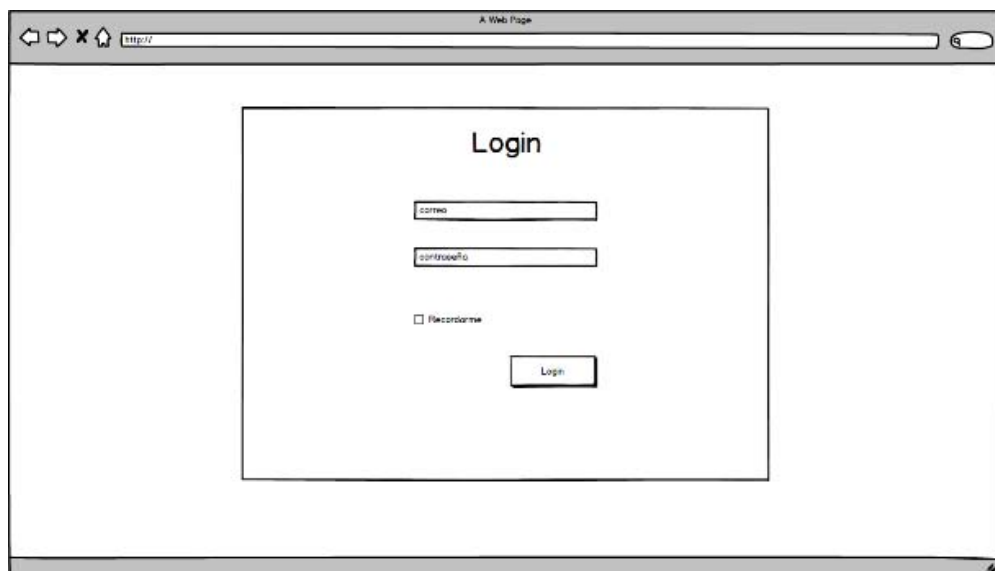


Ilustración 13: Prototipo login

Un usuario que no se haya registrado, podrá hacerlo introduciendo su correo electrónico y una contraseña.



Ilustración 14: Prototipo registro

Ilustración 14: Prototipo registro. Se muestra una ventana de navegador con la URL 'http://'. El contenido principal es un formulario centrado con el título 'Registro'. El formulario contiene tres campos de texto: 'correo', 'contraseña' y 'confirmar contraseña', seguidos de un botón 'Registro'.

Ilustración 14: Prototipo registro

El layout se compartirá en todas las siguientes pantallas y en él existe un menú lateral con información del usuario, un menú superior con las diferentes opciones, un buscador de amigos y la opción de salir.

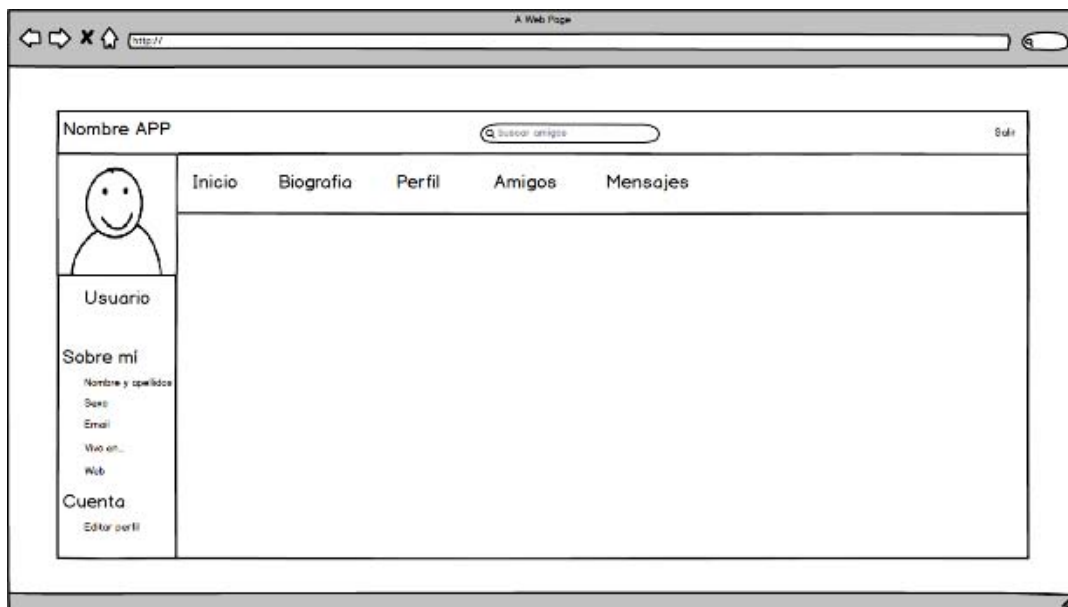


Ilustración 15: Prototipo layout

Ilustración 15: Prototipo layout. Se muestra una ventana de navegador con la URL 'http://'. El layout incluye un encabezado con 'Nombre APP', un buscador de amigos y un botón 'Salir'. A la izquierda hay un menú lateral con un icono de usuario, 'Usuario', 'Sobre mí' (con subopciones: Nombre y apellidos, Sexo, Email, Vivo en..., Web) y 'Cuenta' (con subopción: Editor perfil). El menú superior contiene 'Inicio', 'Biografía', 'Perfil', 'Amigos' y 'Mensajes'. El área principal es un espacio vacío para el contenido.

Ilustración 15: Prototipo layout

En la interfaz de inicio se mostrarán los posts del usuario que ha iniciado sesión junto con los de sus amigos, todos ordenados cronológicamente. En esta interfaz se podrá publicar un post y comentar los posts de otros usuarios.

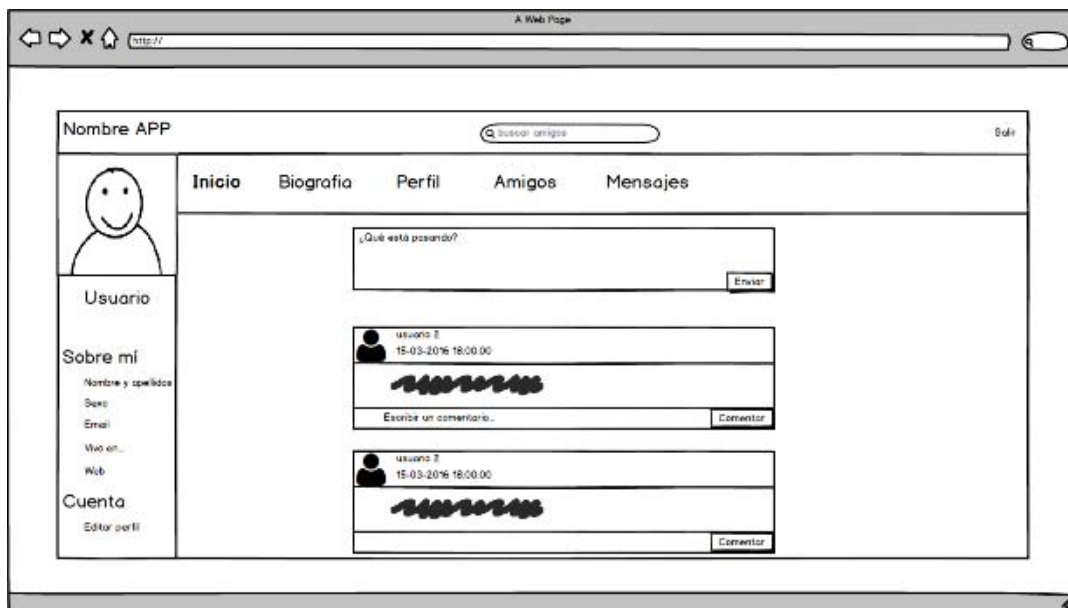


Ilustración 16: Prototipo inicio

La interfaz de biografía es muy similar a la anterior con la diferencia que en esta solamente aparecen los posts del usuario que ha iniciado sesión

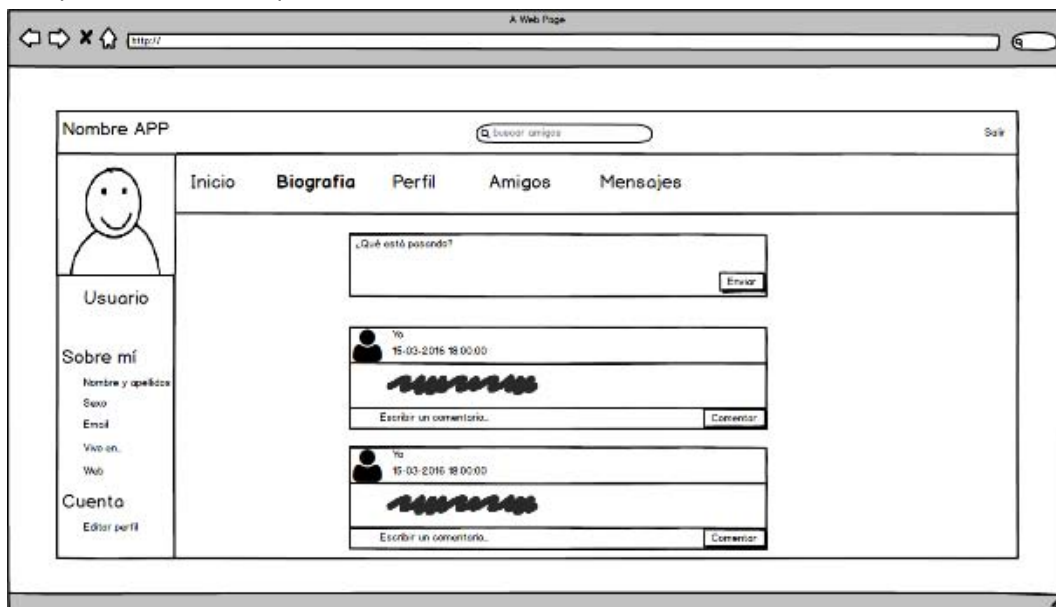


Ilustración 17: Prototipo biografía

En la interfaz de perfil, el usuario que ha iniciado sesión podrá actualizar su información y su foto de perfil.

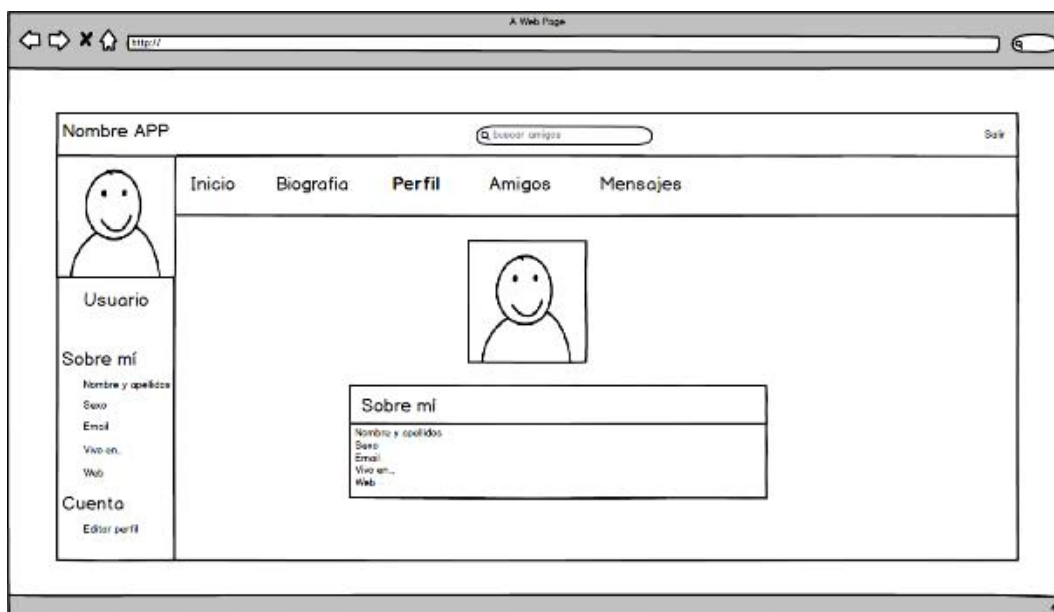


Ilustración 18: Prototipo perfil

Cuando el usuario utilice el buscador o acceda al menú amigos, se mostrarán los resultados en la siguiente pantalla. En cada bloque se mostrará al usuario con un breve resumen de su actividad y además un botón agregar o eliminar.

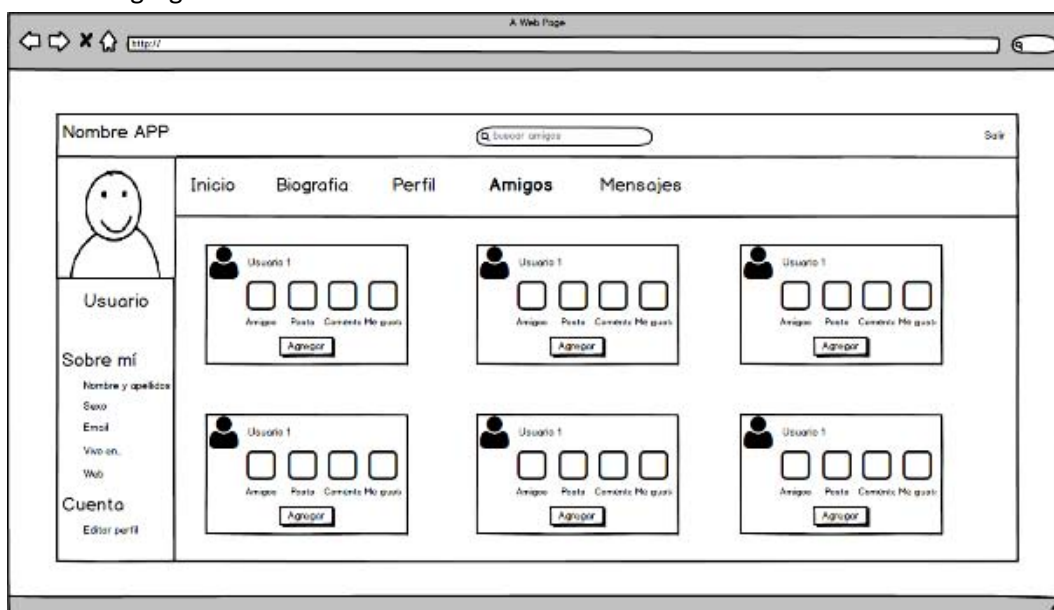


Ilustración 19: Prototipo amigos

En la interfaz de mensajes, se mostrará la lista de usuarios y al seleccionar un usuario, se cargarán los mensajes que se han enviado.



Ilustración 20: Prototipo mensajes

IMPLEMENTACIÓN

La **disciplina de implementación** es el flujo de trabajo, incluyendo actividades, trabajadores y documentación, cuyo principal propósito es definir la organización del código e implementar el sistema en términos de componentes.

ESTRUCTURA DEL PROYECTO

Para comenzar el desarrollo de la aplicación se ha creado un proyecto ASP.NET MVC con login.

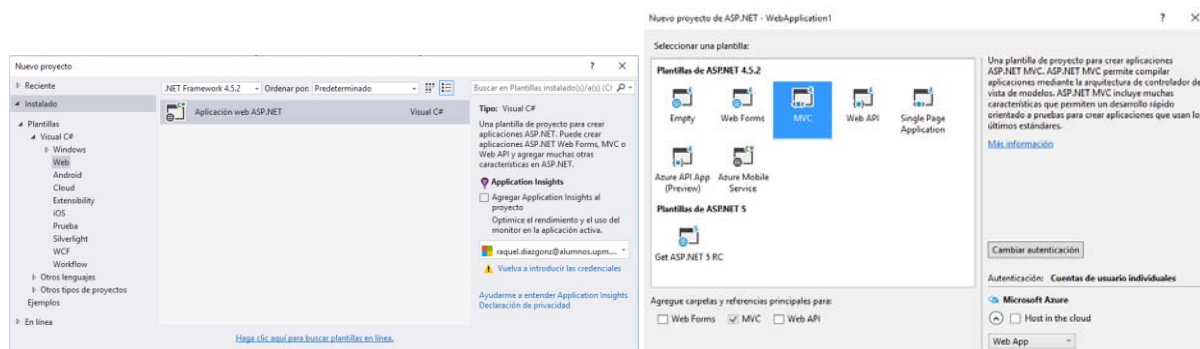


Ilustración 21: Creación del proyecto

En la siguiente imagen se puede ver la estructura completa del proyecto:

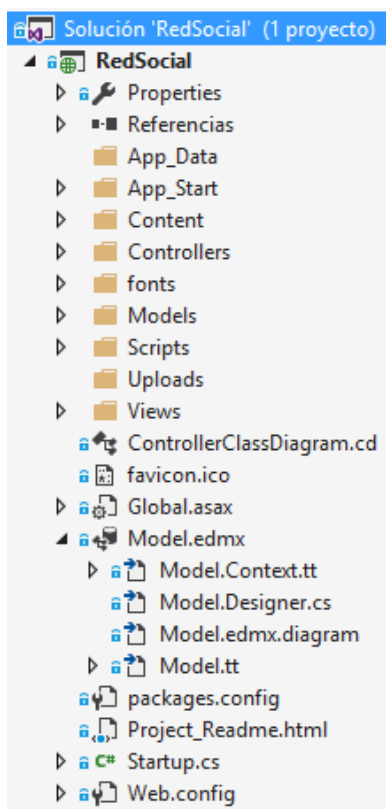
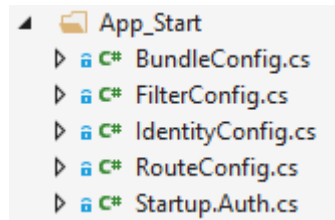


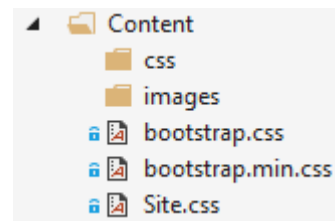
Ilustración 22: Estructura proyecto

App_Data puede contener archivos de datos de aplicación como LocalDB, archivos .mdf, archivos XML y otros datos de los archivos relacionados.

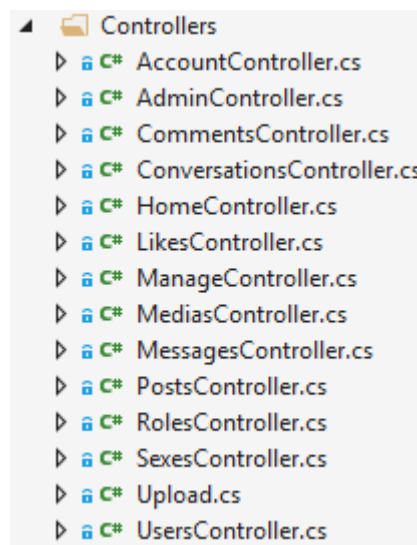
App_Start contiene los archivos de configuración que serán ejecutados cuando se inicia la aplicación. MVC 5 incluye por defecto BundleConfig.cs, FilterConfig.cs y RouteConfig.cs.



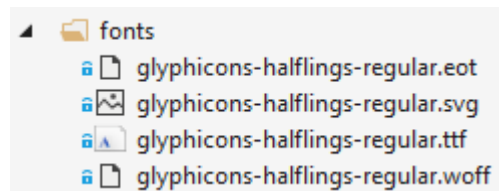
Content es la carpeta de contenido que contiene archivos estáticos (CSS, imágenes, JS). MVC 5 aplicación incluye bootstrap.css, bootstrap.min.css y Site.css por defecto.



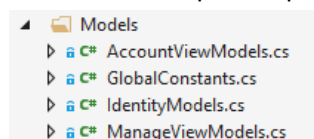
Controllers es la carpeta de controladores y contiene los archivos para los controladores. MVC requiere que el nombre de todos los archivos de los controladores acabe en "Controller".



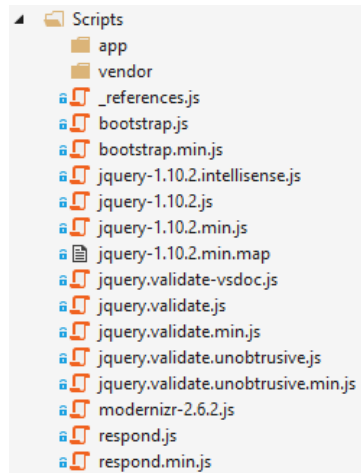
Fonts es la carpeta de fuentes que contiene archivos de fuente personalizados para la aplicación.



Models contiene los modelos que serán utilizados por la aplicación para manipular los datos.

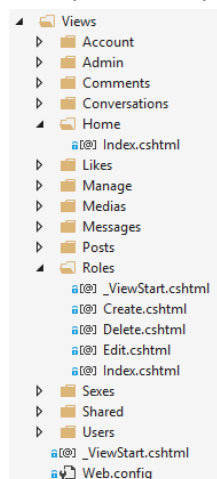


Scripts contiene los archivos JavaScript o VBScript para la aplicación. MVC 5 incluye archivos javascript para arranque, jQuery 1,10 y modernizador de forma predeterminada.



Uploads es la carpeta que se utiliza para guardar las imágenes que los usuarios cargan una vez iniciada la aplicación.

Views contiene archivos HTML para la aplicación. Son archivos .cshtml (razor) escritos en HTML y C#. Cada controlador tendrá asociado su correspondiente paquete de vistas.



Global.asax permite escribir código que se ejecuta en respuesta a eventos de nivel de aplicación.

Packages.config es el archivo administrado por NuGet para realizar un seguimiento de lo que los paquetes y las versiones que se han instalado en la aplicación.

Web.config contiene configuraciones de nivel de aplicación.

PROCESO DE DESARROLLO

Como se ha comentado anteriormente, el desarrollo del proyecto se ha llevado a cabo siguiendo la **metodología Scrum**, y para ello, se ha hecho uso de la herramienta **SeeNowDo** disponible en www.seenowdo.com



Esta herramienta es totalmente gratuita y permite organizar el proyecto por *sprints* añadiendo a cada uno una serie de historias de usuario. Además, a cada historia de usuario se le puede asignar un conjunto de tareas priorizadas y estimadas (*Sprint Backlog*), lo que será útil para poder obtener después los diagramas *Burn Down*.

Debido a que en el desarrollo de este proyecto solo estaba implicada una persona, se ha decidido reducir la duración de los *sprints*. La teoría de la metodología ágil de *Scrum*, dice que un sprint debe durar como máximo 4 semanas, pero en la práctica se ha decidido realizar *sprints* con 1 semana de duración en los que se empezaba un viernes y se acababa un jueves. Esto facilitaba bastante la organización del proyecto con el trabajo.

A continuación se puede ver en el calendario la duración del proyecto y su organización por *sprints*.

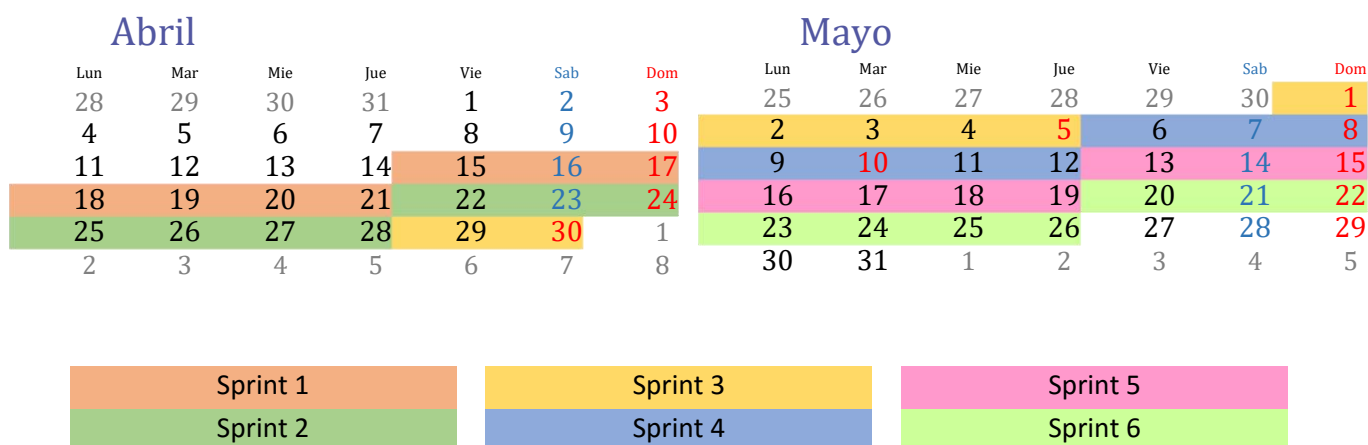


Ilustración 23: Calendario sprints

- **Sprint 1**

En la siguiente imagen se puede ver las historias de usuario y las tareas asociadas. Es el primer *sprint*, por lo que es necesario dedicar este tiempo a estudiar las tecnologías y a preparar el proyecto.

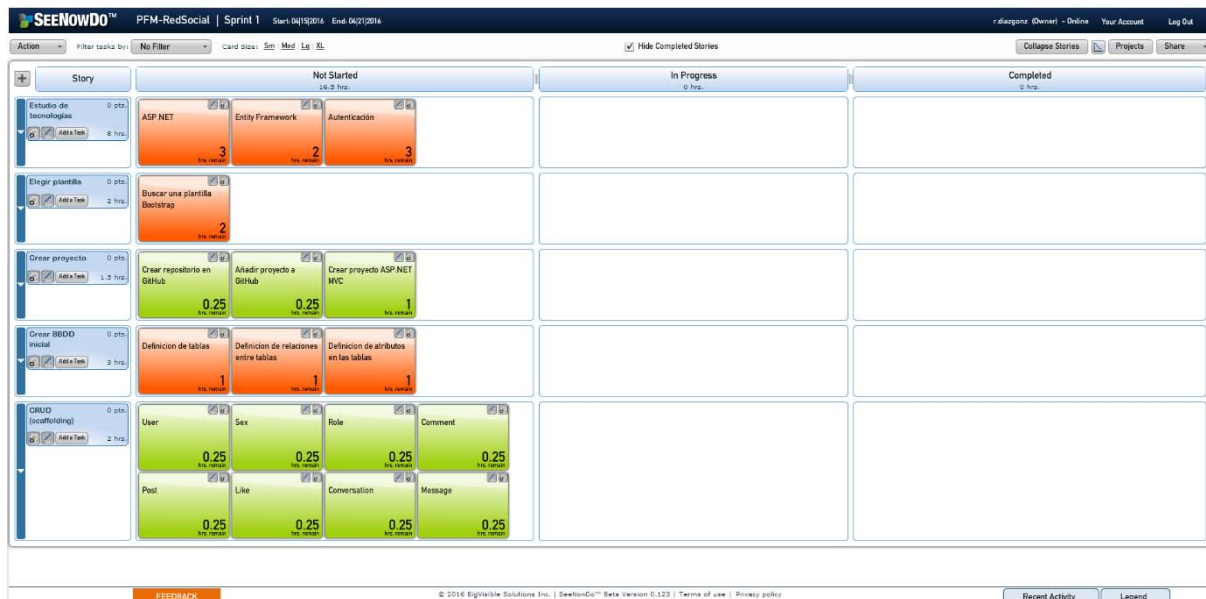


Ilustración 24: Backlog sprint 1

En el gráfico *BurnDown* se puede ver que todas las tareas planificadas en el sprint se completaron, aunque al principio había un poco de retraso.



Ilustración 25: Burn down chart sprint 1

• Sprint 2

Una vez que estaba definido el proyecto y las tecnologías, había que integrar la plantilla elegida y diseñar las vistas que posteriormente formarían la interfaz de la red social. También se dedica unas horas a integrar la autenticación que proporciona .NET.

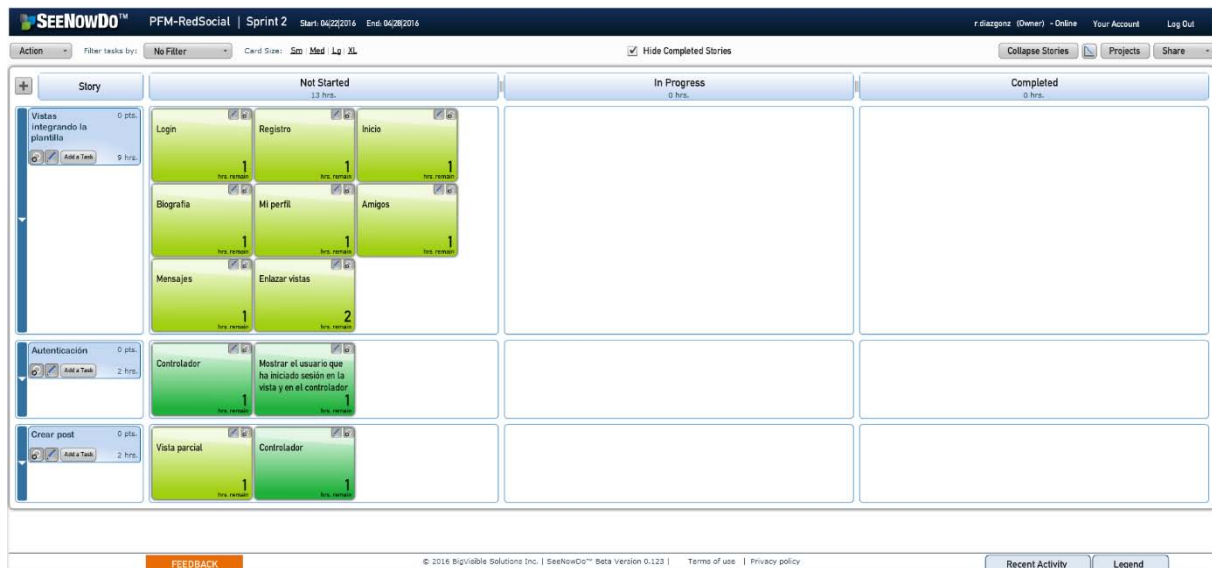


Ilustración 26: Backlog sprint 2

Este sprint se completó un día antes de lo previsto, por lo que no hubo ningún problema.



Ilustración 27: Burn down chart sprint 2

- **Sprint 3**

Este sprint es el que mayor carga de funcionalidad tiene porque en él se desarrollan la mayor parte de los requisitos definidos para la red social.

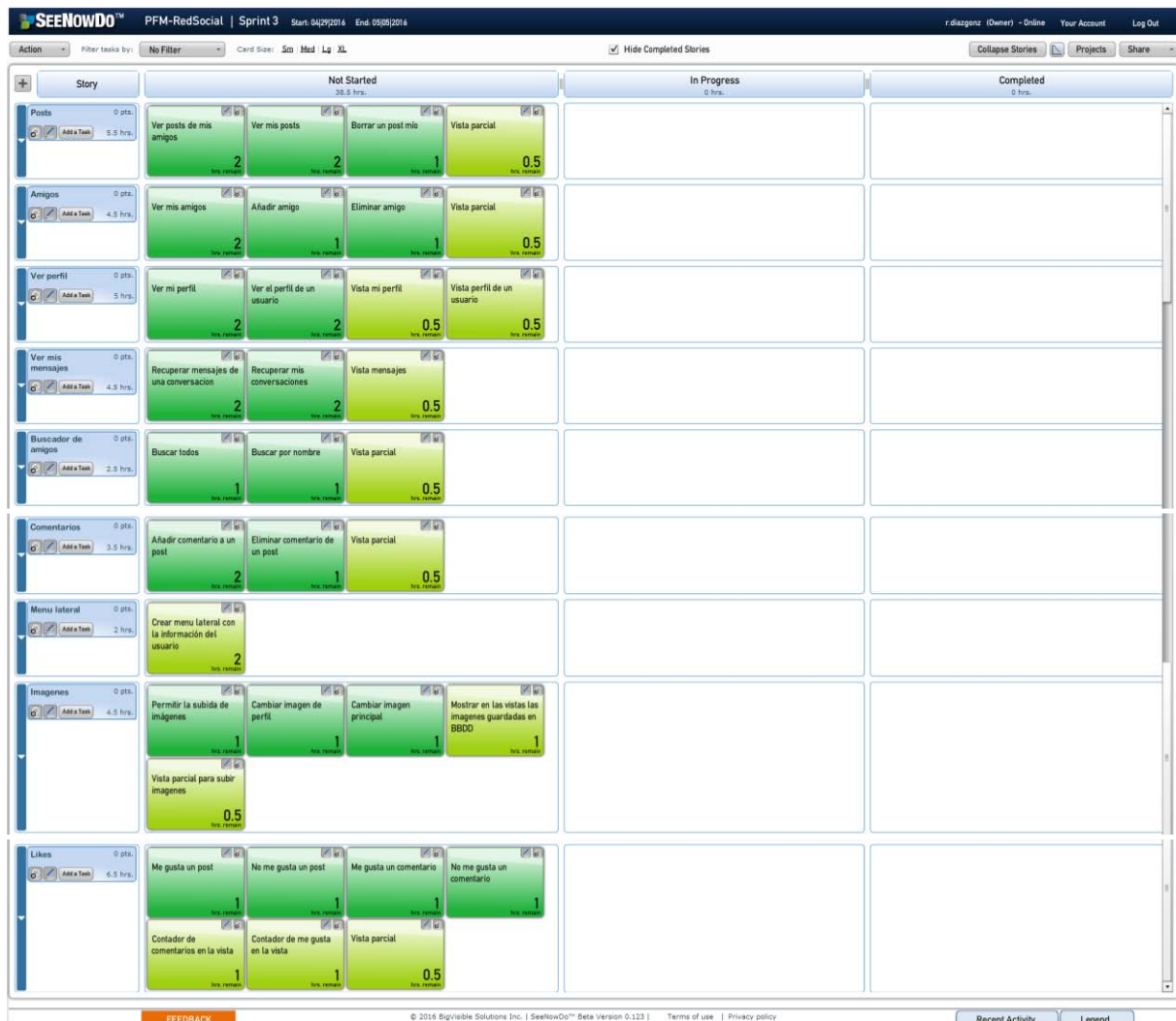


Ilustración 28: Backlog sprint 3

Pero aun teniendo mucha carga de funcionalidad, todos los requisitos fueron desarrollados en 4 días. Esto es debido a que el desarrollo con *.NET* resultaba bastante fácil y porque las operaciones *crud* mediante *scaffolding* ya estaban desarrolladas en el *sprint* 1.



Ilustración 29: Burn down chart sprint 3

- **Sprint 4**

El cuarto sprint, con menos carga de trabajo que el anterior, se centraba en el envío de mensajes, la estandarización de iconos y la subida de imágenes y vídeos.

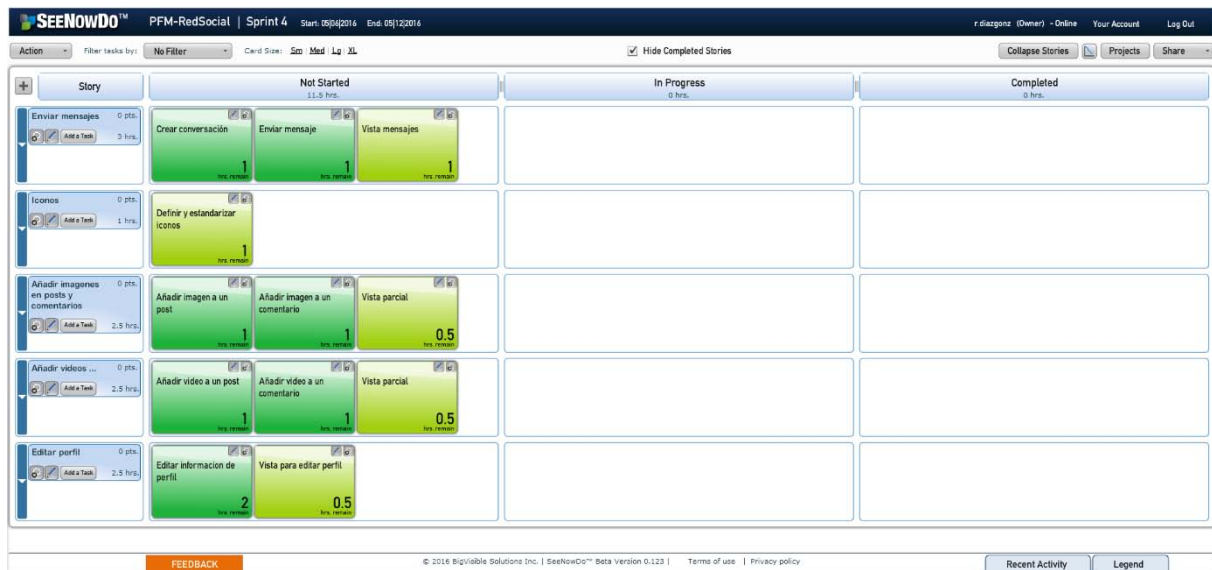


Ilustración 30: Backlog sprint 4

Todas las tareas fueron realizadas en 4 días, y también se terminó antes de lo previsto debido a que el desarrollo de la subida de imágenes, que se consideraba bastante complicado, en realidad, no dio ningún tipo de problema.



Ilustración 31: Burn down chart sprint 4

• Sprint 5

En el quinto sprint, también había bastante carga de funcionalidad debido a que se pretendía realizar la parte de administración. Por tanto, era necesario crear todas las vistas de administración, además de todos los controladores. También se reservó una parte del tiempo del sprint para corregir pequeños errores que habían ido surgiendo de los *sprints* anteriores.

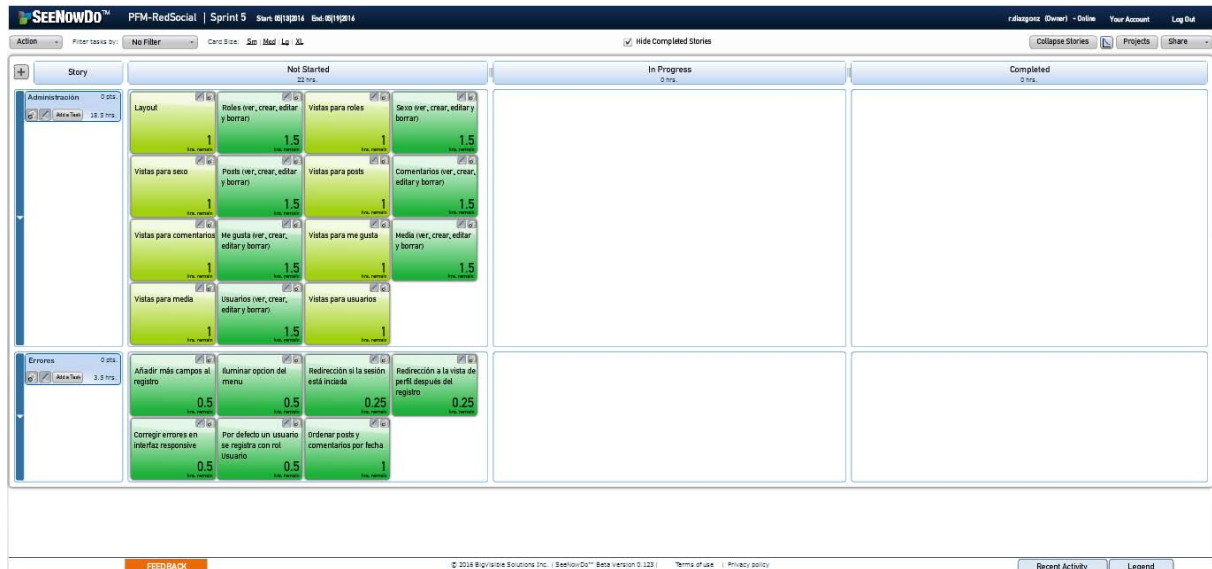


Ilustración 32: Backlog sprint 5

En el gráfico se puede ver que sobraron 2 días debido a que el desarrollo era bastante rápido y no surgió ningún tipo de problema.

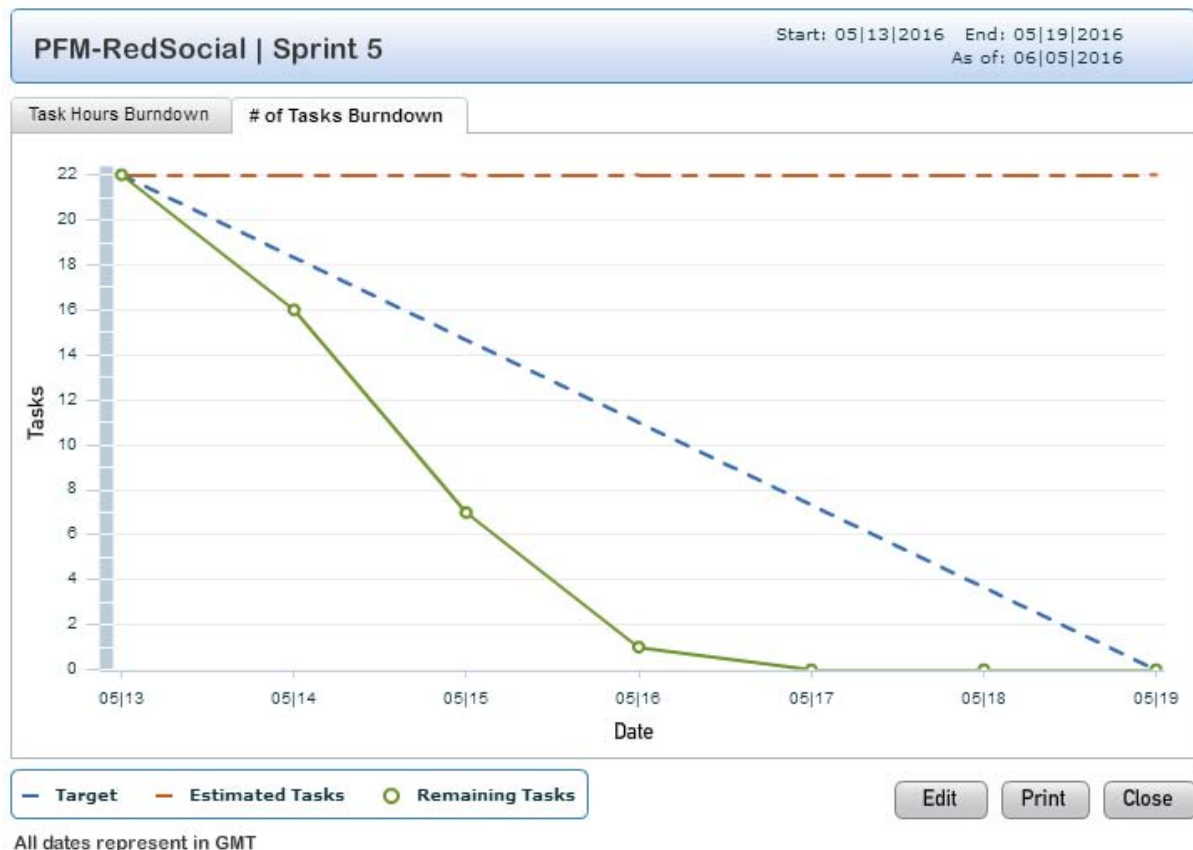


Ilustración 33: Burn down chart sprint 5

- **Sprint 6**

Por último, el sexto *sprint* se dedica a realizar mejoras, corrección de errores y refactorizaciones.

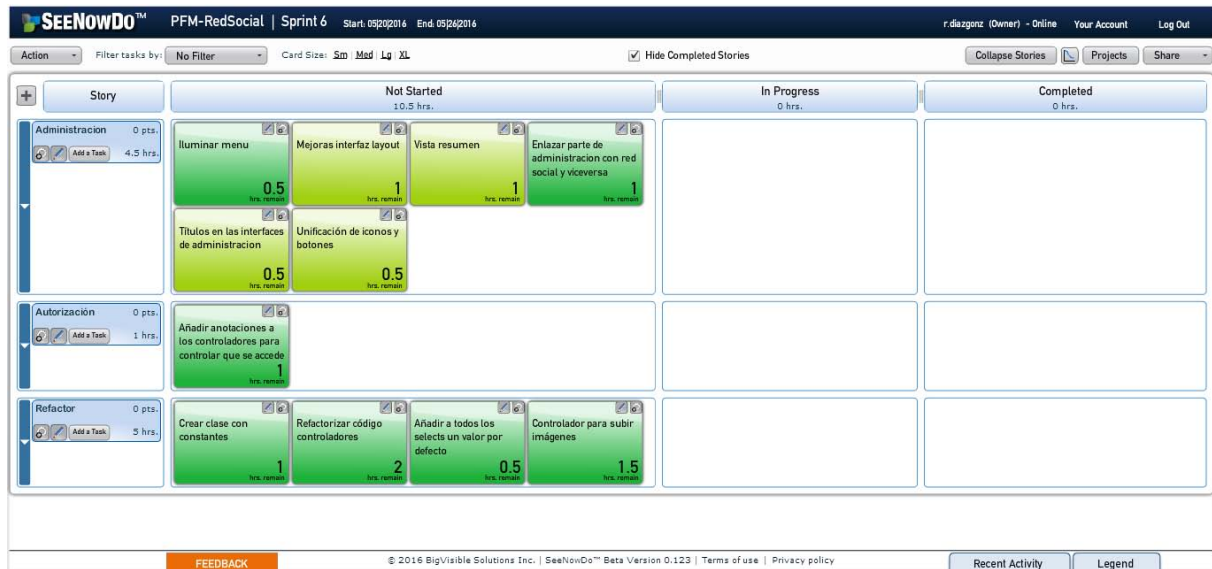


Ilustración 34: Backlog sprint 6

Este *sprint* también se terminó a tiempo aunque hubo una parada de 3 días en los que no se pudo realizar ningún tipo de funcionalidad.

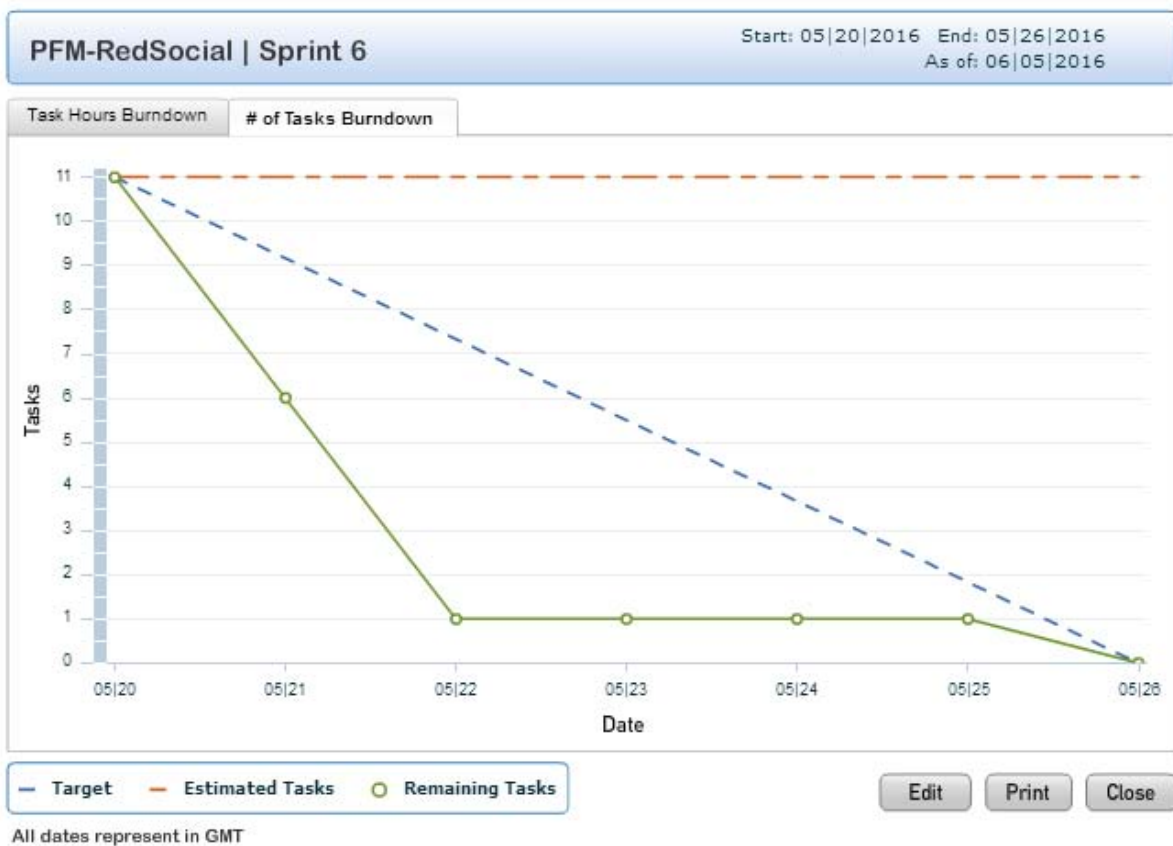


Ilustración 35: Burn down chart sprint 6

La estimación del tiempo de trabajo del desarrollo del proyecto son 112 horas, y se puede resumir en la siguiente gráfica:

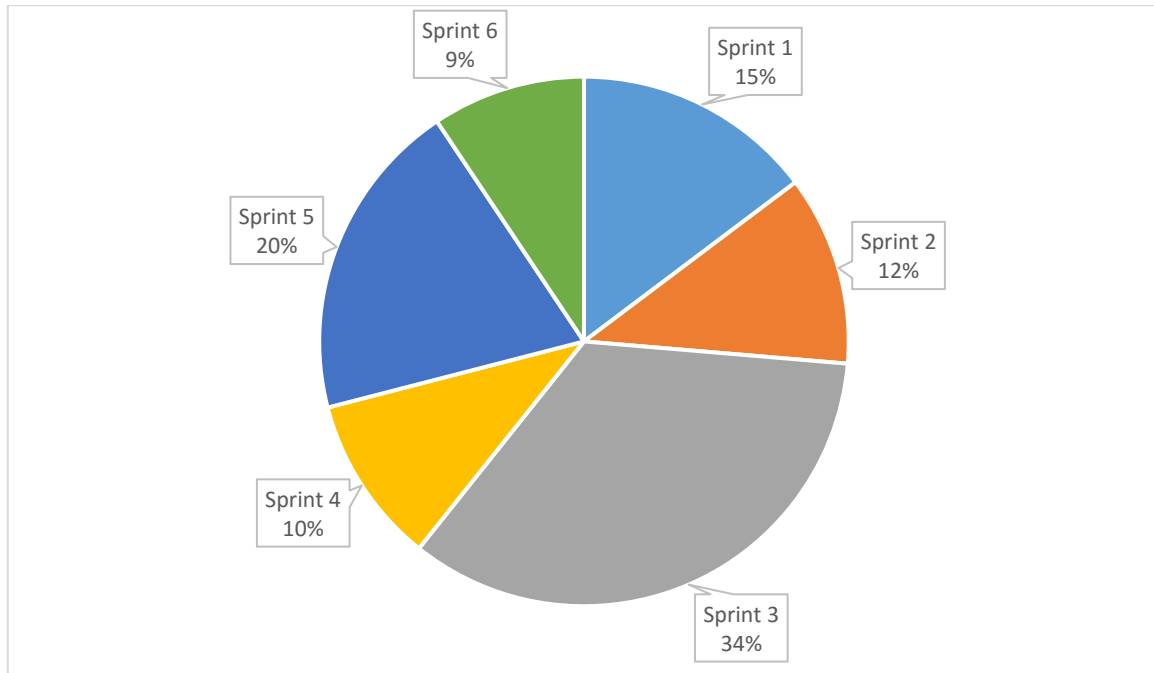


Ilustración 36: Gráfico tiempo de desarrollo

EVOLUCIÓN DE LA INTERFAZ DE USUARIO

Se han insertado datos de muestra para poder ver un ejemplo funcional, de tal forma que se puede iniciar sesión con cualquier usuario y con la contraseña: Social_2016.

En el siguiente diagrama se puede ver la relación que existe entre todos los **usuarios** dados de alta.

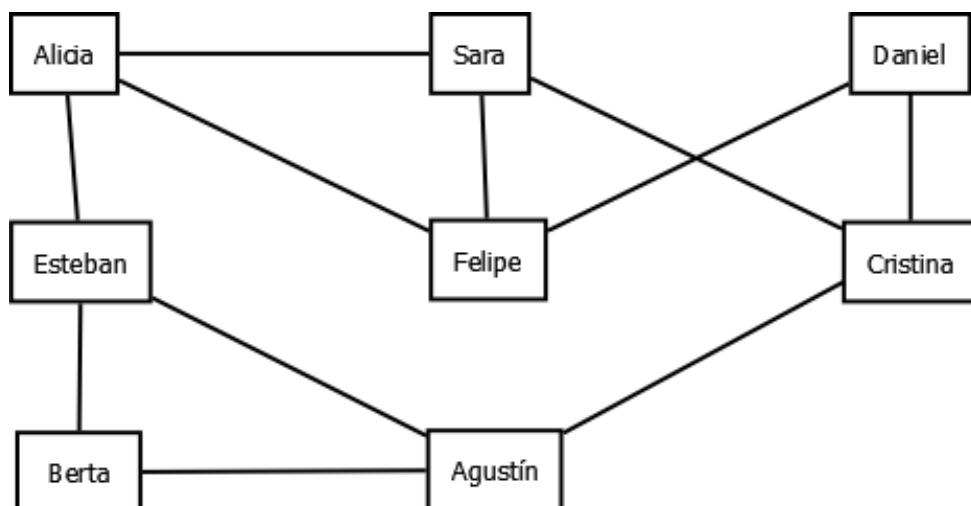


Ilustración 37: Grafo usuarios

Los **iconos** que se utilizan en toda la aplicación son los que se pueden ver en la siguiente tabla:

Inicio	 fa-home	Enviar/guardar	 fa-send-o
Biografía	 icon-ship-wheel	Comentar	 fa-comments-o
Perfil	 fa-user	Eliminar	 fa-trash-o
Amigos	 fa-users	Nombre de usuario y nombre completo	 fa-user
Mensajes	 md-chat	Sexo	 fa-venus-mars
Comentario	 fa-comments	Email	 fa-envelope-o
Me gusta	 fa-heart	Donde vivo	 fa-map-marker
Buscar	 fa-search	Web	 fa-globe

Ilustración 38: Tabla iconos

A continuación se muestra el desarrollo de la red social mediante capturas de la interfaz.

- **Inicio**

Pantalla principal al entrar en la aplicación que contiene los enlaces para registrarse o iniciar sesión.



Ilustración 39: Interfaz inicio aplicación

- **Login**

Cuando un usuario ya se ha dado de alta, puede entrar a la aplicación mediante su usuario y su contraseña.

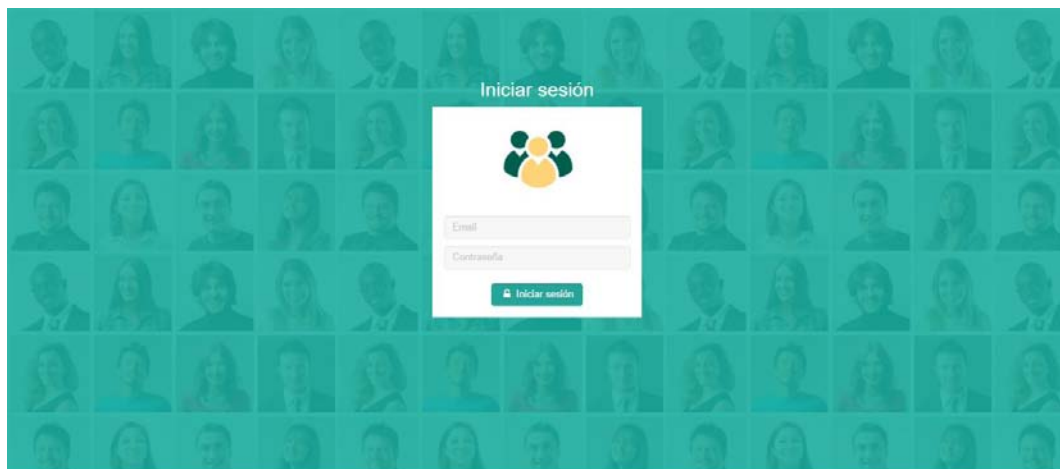
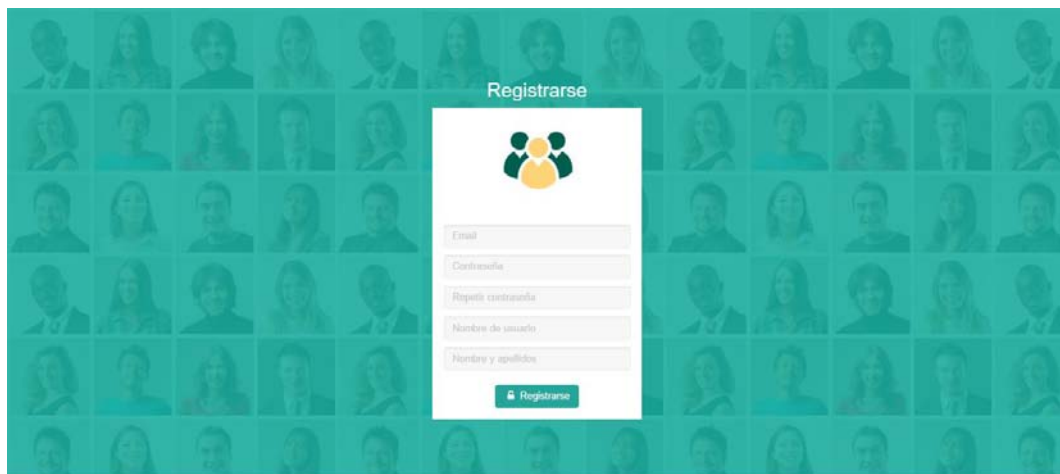


Ilustración 40: Interfaz iniciar sesión

○ Registro

Cuando un usuario necesita registrarse, accederá a esta pantalla y cumplimentando el formulario, tendrá acceso a la aplicación

The image shows a registration form titled 'Registrarse' centered on a teal background with a grid of small, faded user avatars. The form itself is white and contains the following fields: 'Email', 'Contraseña', 'Repetir contraseña', 'Nombre de usuario', and 'Nombre y apellidos'. Below these fields is a green button labeled 'Registrarse' with a small icon of a person.

Proyecto Fin de Máster - Raquel Díaz González - © 2016

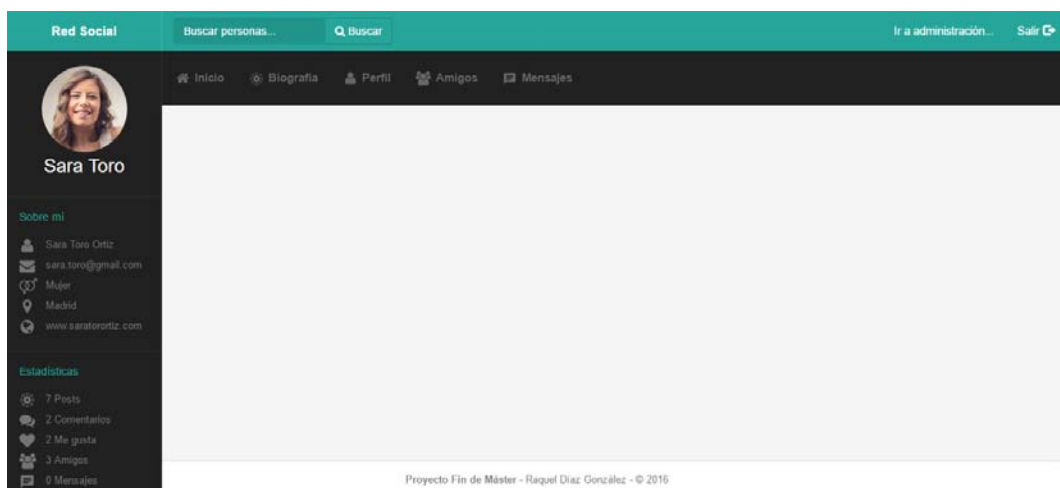
Ilustración 41: Interfaz registro

• Red social

○ Layout

Es la parte que comparten todas las pantallas de la aplicación una vez que el usuario ha iniciado sesión. En la siguiente imagen se puen ver las opciones disponibles:

- **Buscador de amigos**
- **Menú lateral (imagen y nombre del usuario que ha iniciado sesión, sobre mí y estadísticas)**
- **Menú superior (inicio, biografía, perfil, amigos y mensajes)**
- **Opción para acceder a administración**
- **Salir**



Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 42: Interfaz layout

- Inicio
 - Ver posts (míos y de mis amigos) en orden cronológico

Es la pantalla principal de interacción.

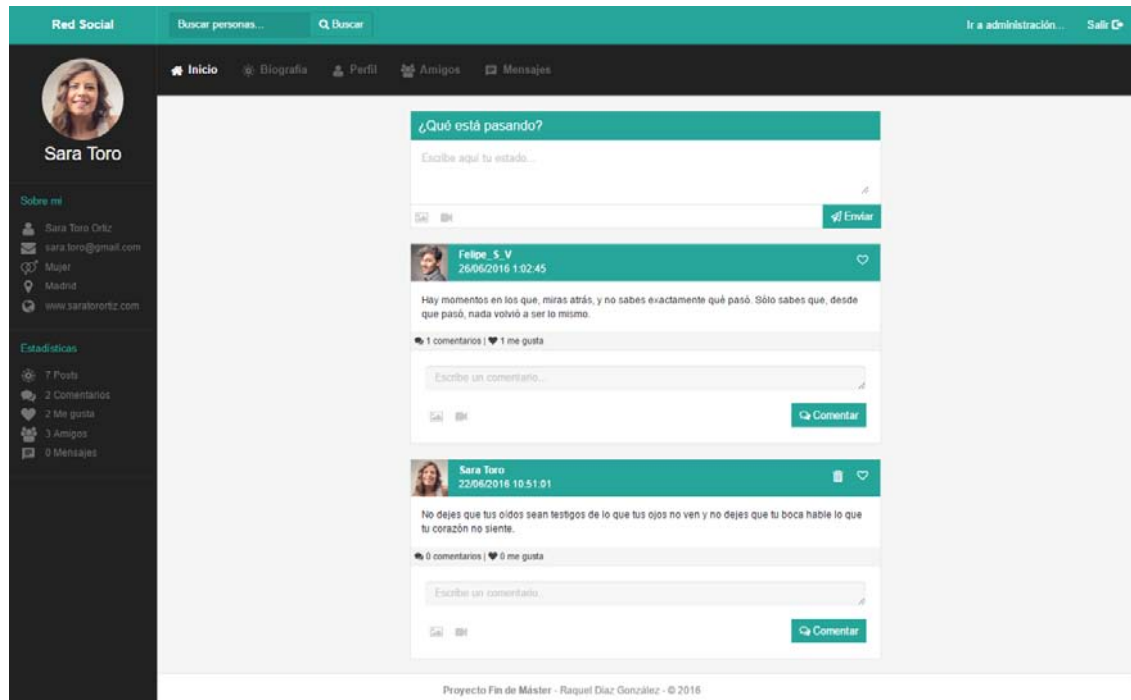


Ilustración 43: Interfaz inicio

- Publicar post

Se podrá publicar un post siempre y cuando el texto no esté vacío.

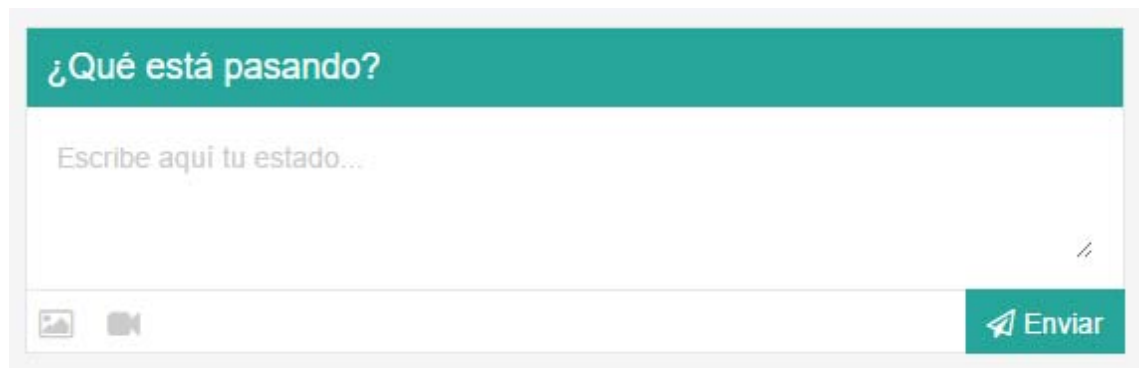
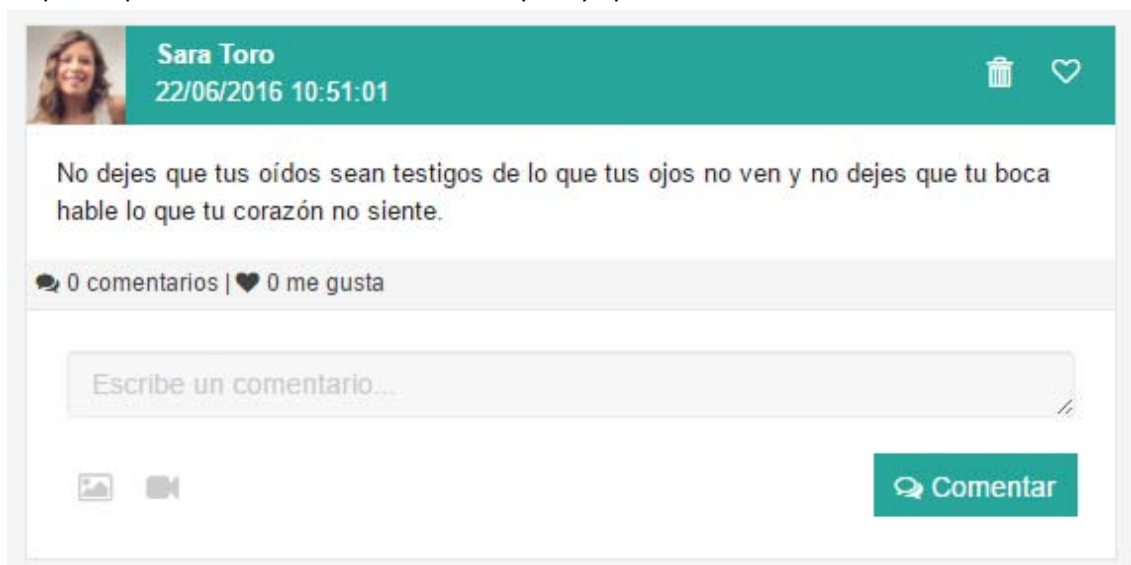


Ilustración 44: Interfaz publicar post

▪ Publicar comentario

Se puede publicar un comentario sobre un post ya publicado



The screenshot displays a social media post by Sara Toro, dated 22/06/2016 at 10:51:01. The post text reads: "No dejes que tus oídos sean testigos de lo que tus ojos no ven y no dejes que tu boca hable lo que tu corazón no siente." Below the text, it shows "0 comentarios" and "0 me gusta". At the bottom, there is a text input field with the placeholder "Escribe un comentario...", a "Comentar" button, and icons for adding images and videos.

Ilustración 45: Interfaz publicar comentario

▪ Publicar imagen en post

Al hacer clic en el icono de la imagen situado abajo a la izquierda, se despliega el selector de imágenes.



Ilustración 46: Interfaz publicar imagen en post

Una vez publicado, se verá así

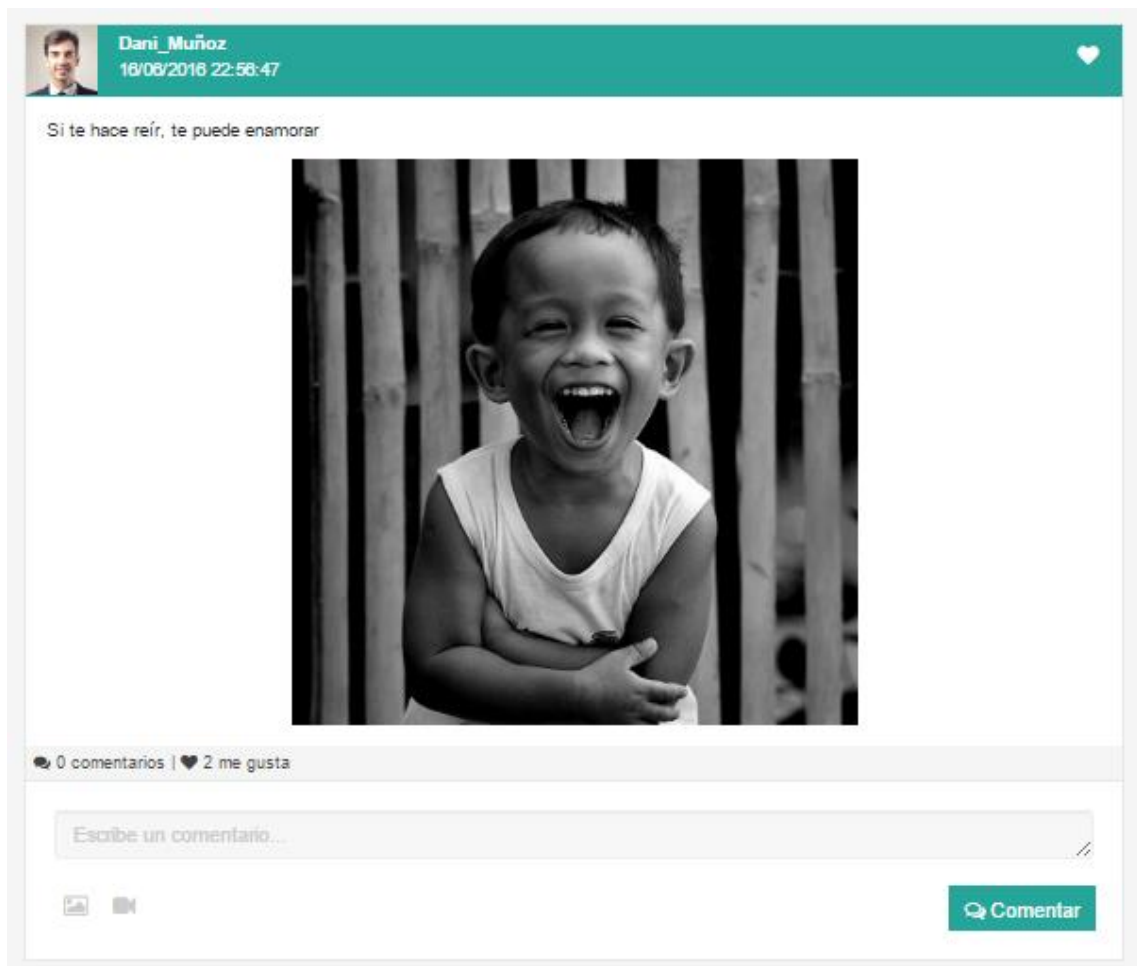
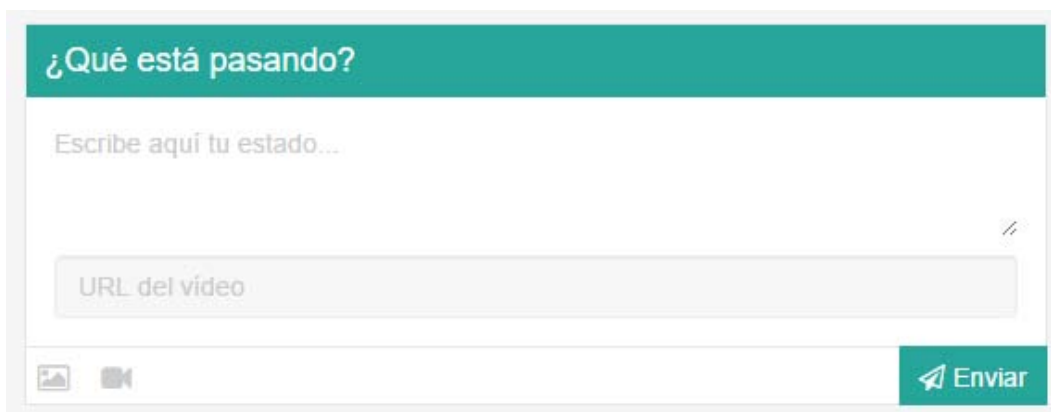


Ilustración 47: Interfaz post con imagen

▪ Publicar vídeo en post



Al hacer clic en el icono del vídeo situado abajo a la izquierda, se despliega el selector de imágenes.



¿Qué está pasando?

Escribe aquí tu estado...

URL del vídeo


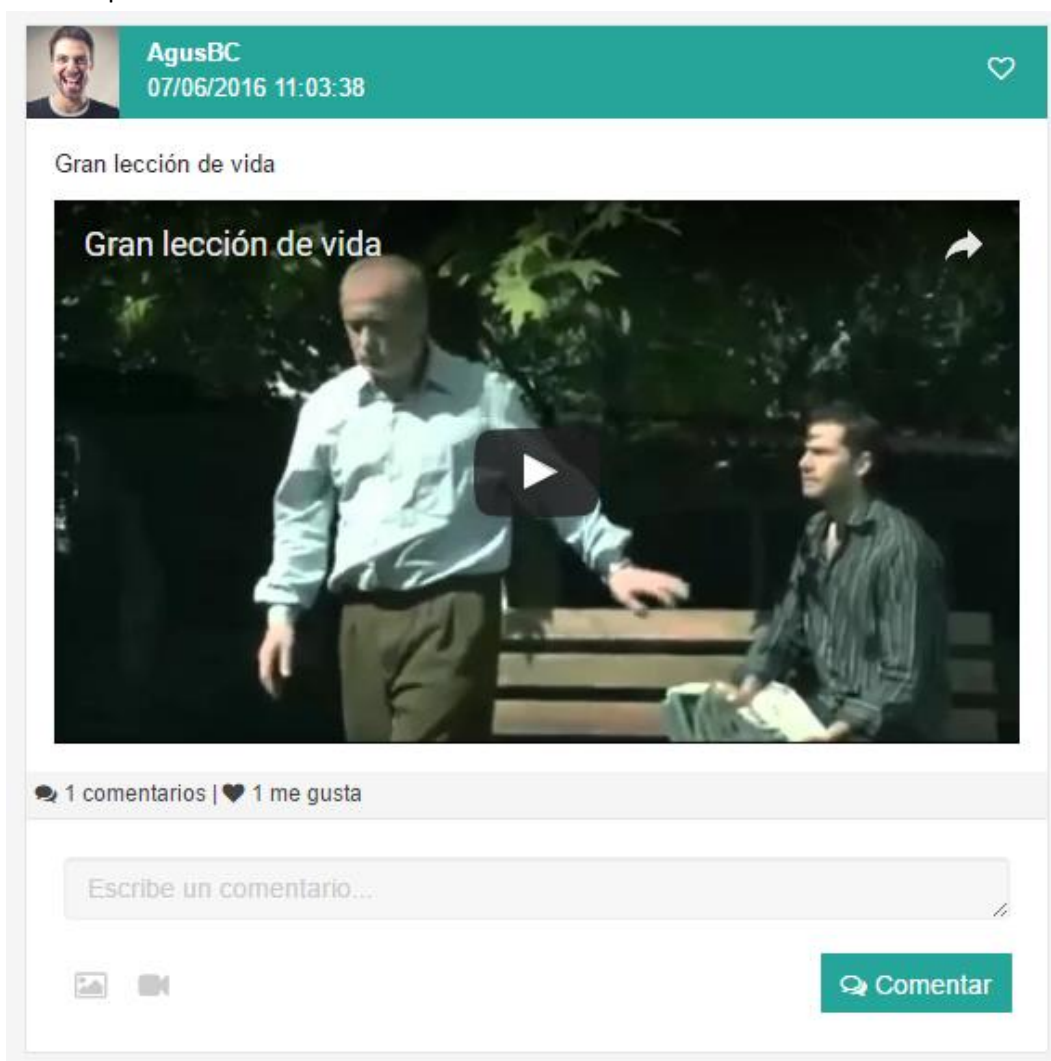


 Enviar


Ilustración 48: Interfaz publicar vídeo en post


Una vez publicado se verá así





 **AgusBC**
07/06/2016 11:03:38 



Gran lección de vida

Gran lección de vida 



 1 comentarios |  1 me gusta

Escribe un comentario...


 Comentar

Ilustración 49: Interfaz post con vídeo

- Eliminar post (mío)
- Eliminar comentario (mío)
- Añadir me gusta en un post
- Añadir me gusta en un comentario
- Eliminar me gusta un post
- Eliminar me gusta un comentario

Se podrán realizar todas las acciones anteriormente indicadas haciendo uso de los iconos definidos en la interfaz

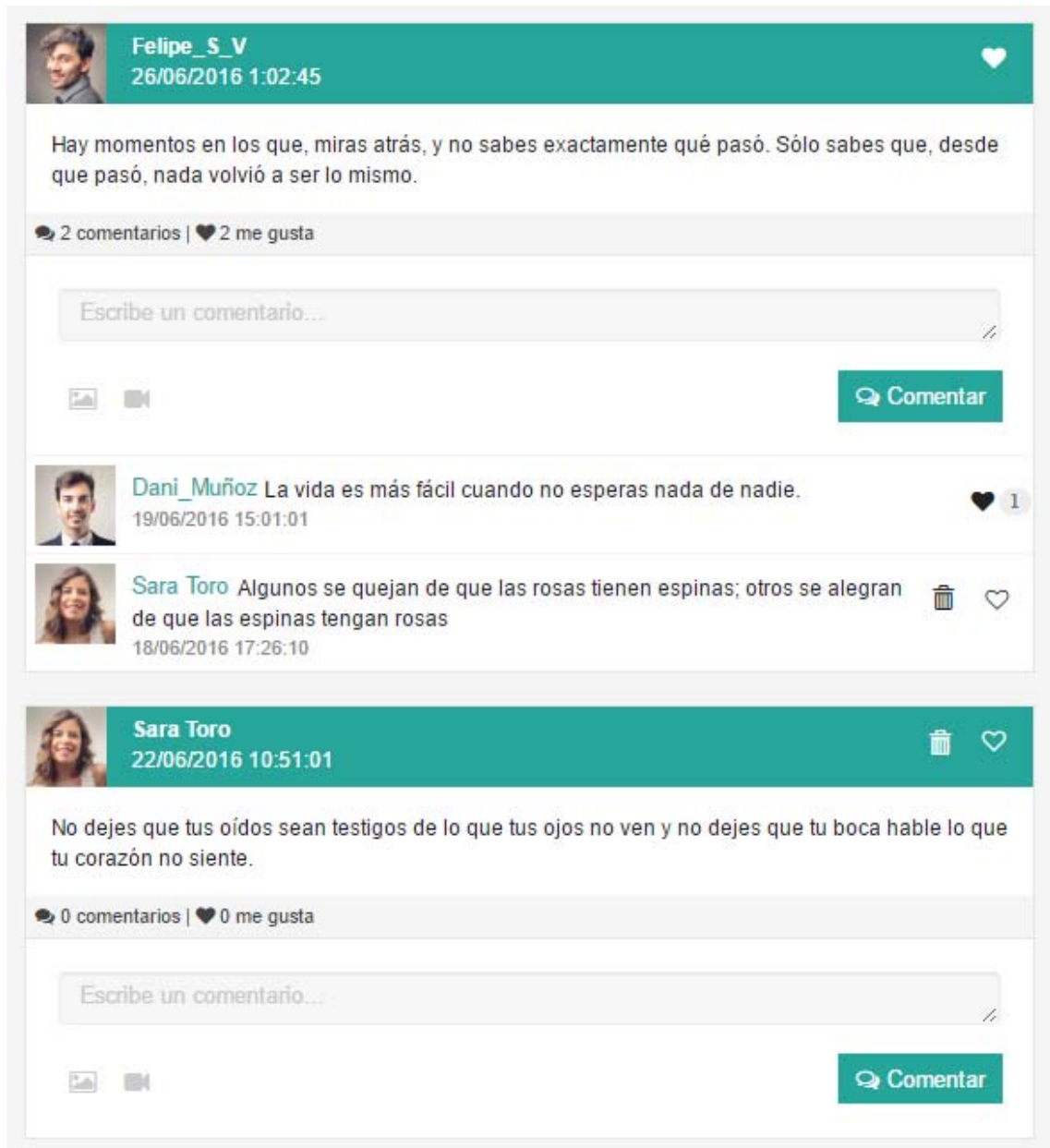


Ilustración 50: Interfaz con varias acciones

- **Biografía**

- **Ver mis posts en orden cronológico**

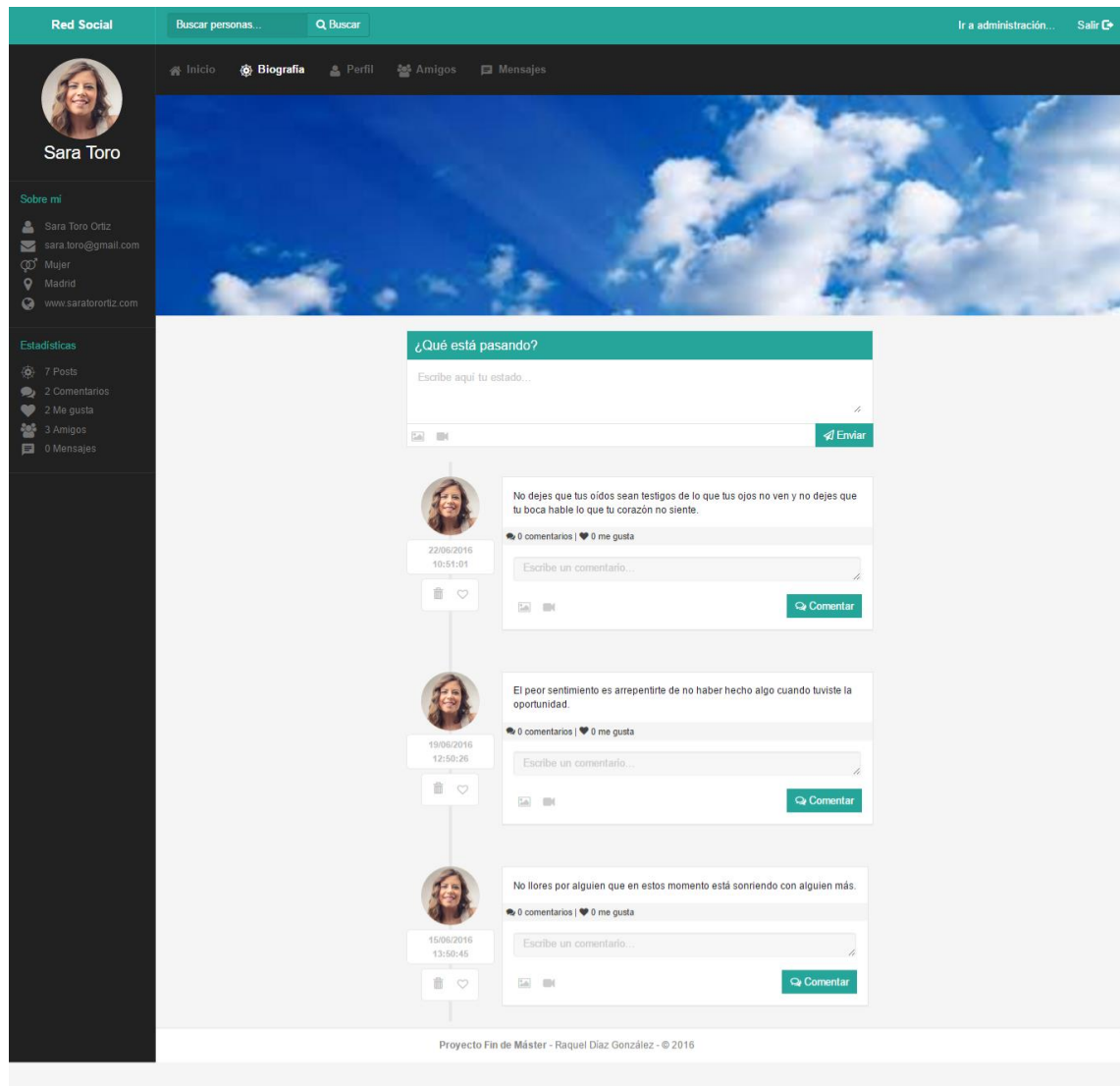


Ilustración 51: Interfaz biografía

- **Publicar post**

Se utiliza el mismo módulo que el de la interfaz de inicio.

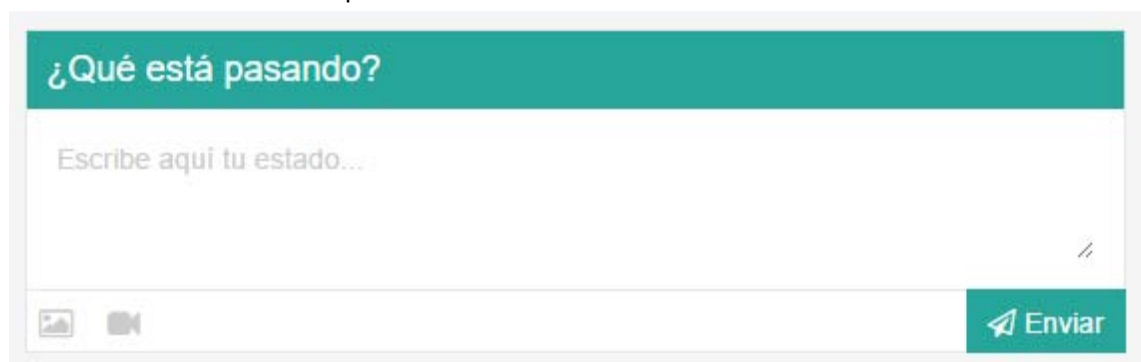


Ilustración 52: Interfaz publicar post en biografía

▪ Publicar comentario

Se puede publicar un comentario sobre un post ya publicado



Ilustración 53: Interfaz publicar comentario en un post de la biografía

▪ Publicar imagen en comentario

Al hacer clic en el icono de la imagen situado abajo a la izquierda, se despliega el selector de imágenes.



Ilustración 54: Interfaz publicar imagen en un comentario de la biografía

▪ Publicar vídeo en comentario

Al hacer clic en el icono del vídeo situado abajo a la izquierda, se despliega el selector de imágenes

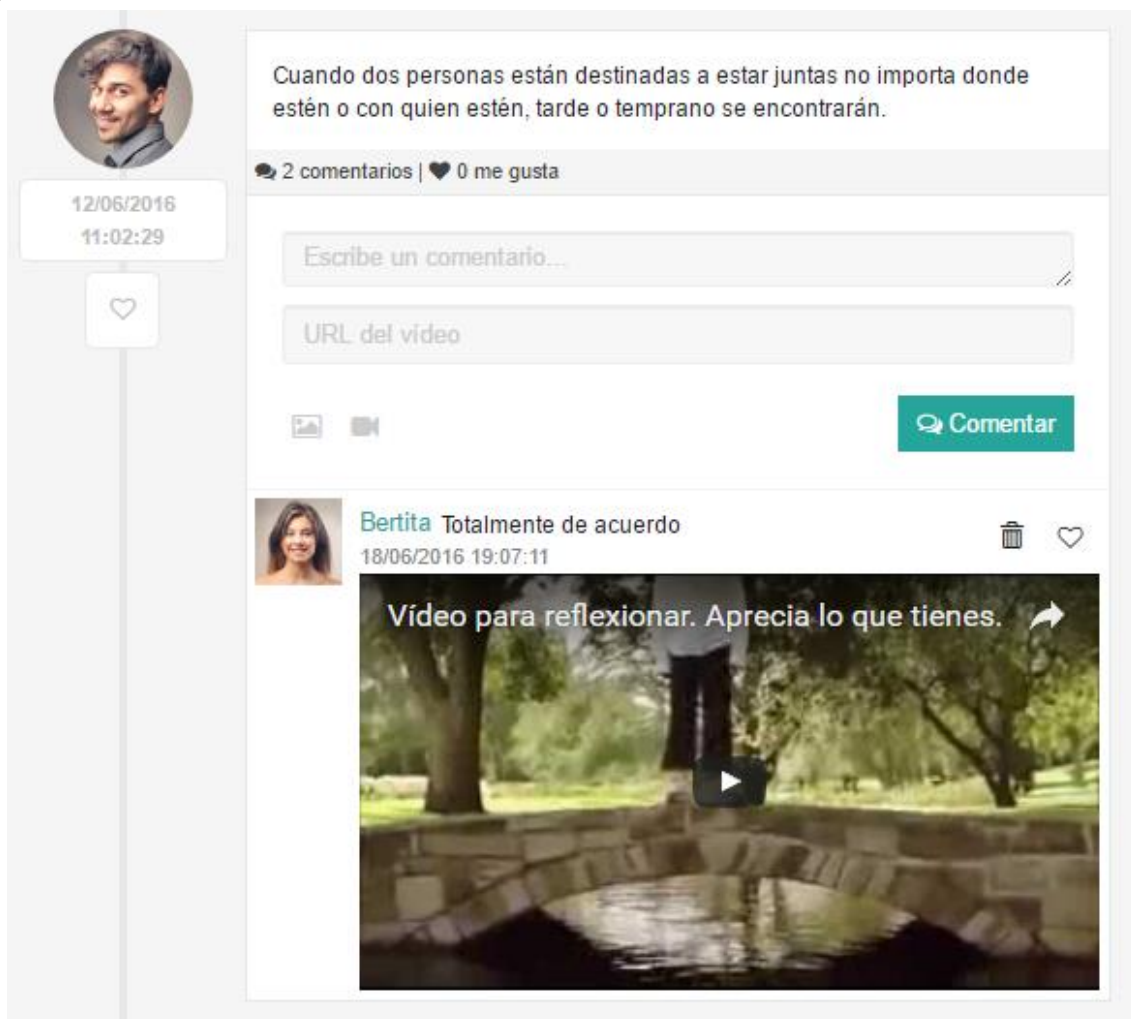


Ilustración 55: Interfaz publicar vídeo en un post de la biografía

- Eliminar post (mío)
- Eliminar comentario (mío)
- Añadir me gusta en un post
- Añadir me gusta en un comentario
- Eliminar me gusta un post
- Eliminar me gusta un comentario

Se podrán realizar todas las acciones anteriormente indicadas haciendo uso de los iconos definidos en la interfaz



Ilustración 56: Interfaz con varias acciones en biografía

- Perfil

- Perfil por defecto al registrarse como nuevo usuario

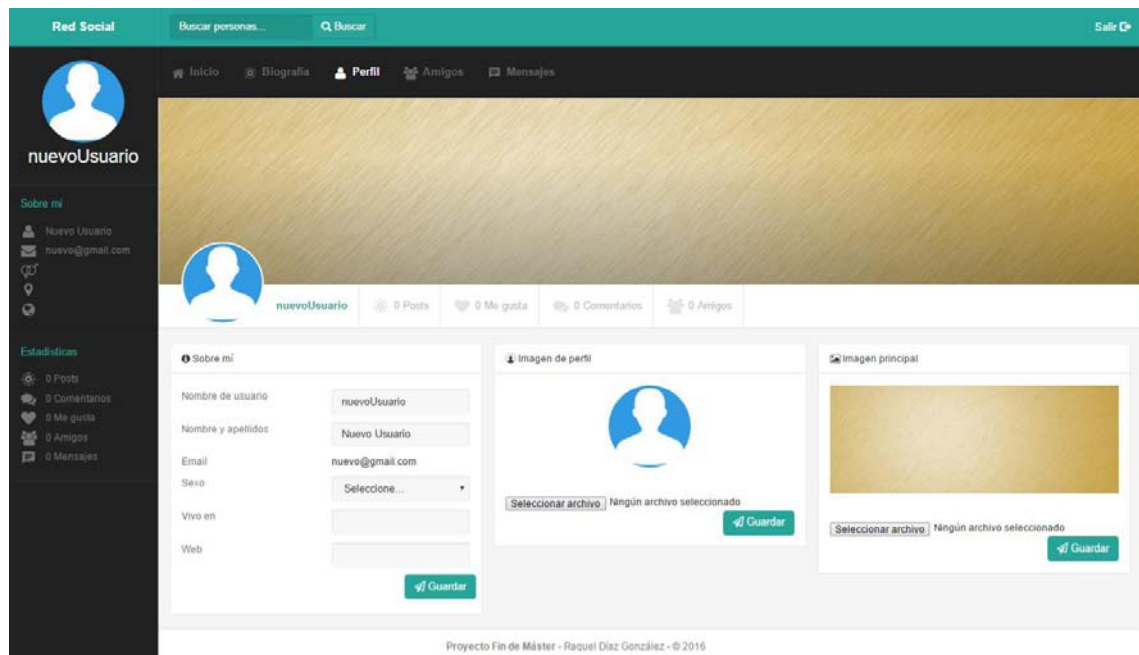


Ilustración 57: Interfaz perfil por defecto

- Ver mi perfil
 - Editar información que se muestra en mi perfil
 - Cambiar imagen de perfil
 - Cambiar imagen principal
 - Ver mis estadísticas

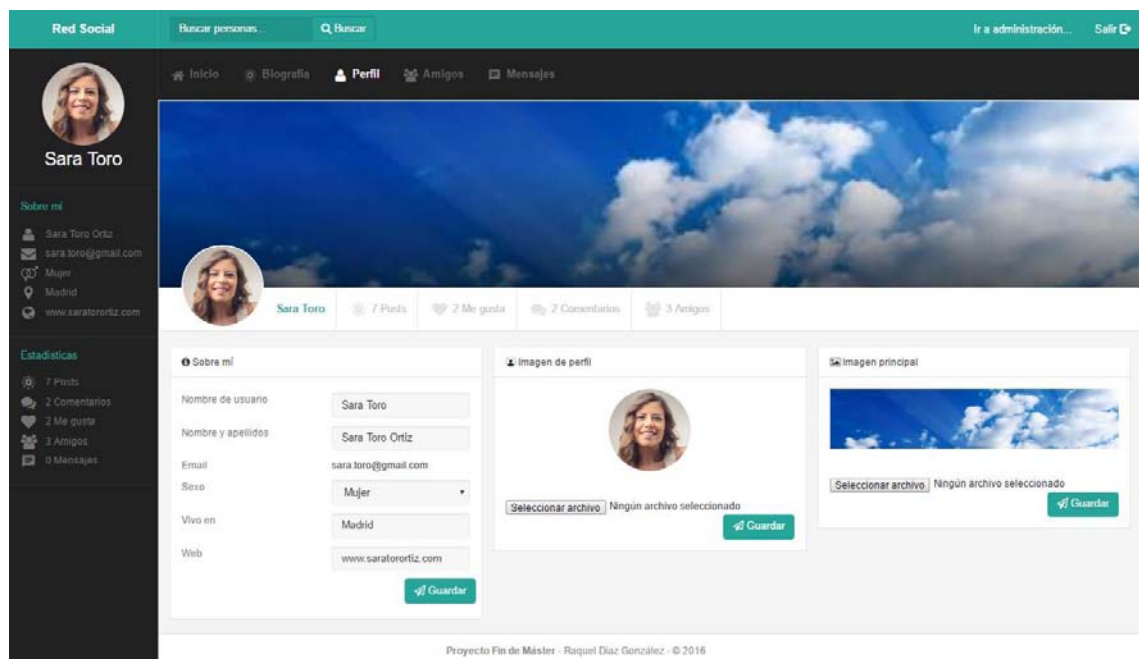


Ilustración 58: Interfaz perfil

○ Amigos

- Ver mis amigos
- Eliminar amigo

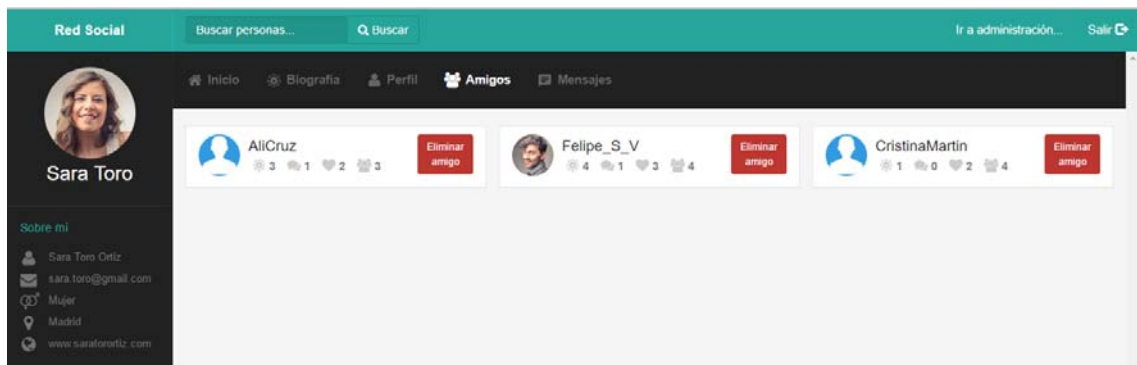


Ilustración 59: Interfaz amigos

- Buscar amigos (filtro buscador)

Se podrán añadir y eliminar amigos utilizando los botones verdes y rojos

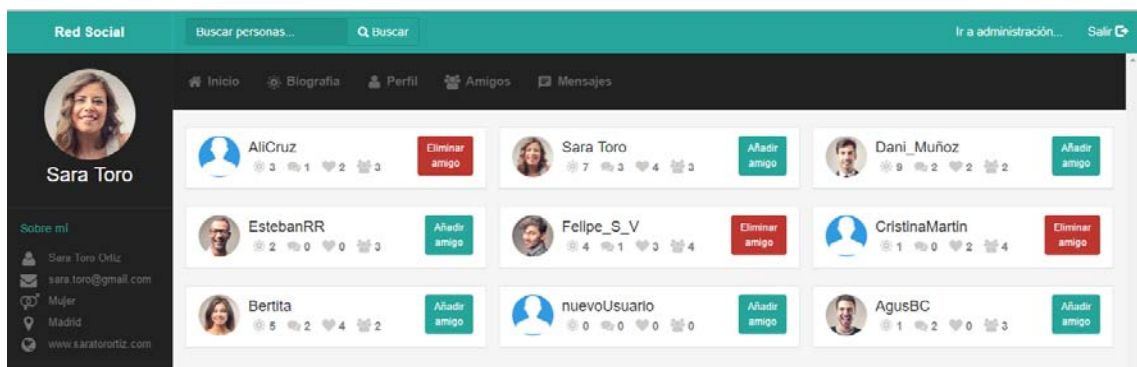


Ilustración 60: Interfaz búsqueda de amigos

- **Mensajes**
 - **Crear conversación**
 - **Enviar mensaje**

En la primera columna de la izquierda se muestra la lista de amigos. Para iniciar una conversación con un amigo, simplemente con seleccionar el icono de la conversación, ésta se creará en la lista de conversaciones. Finalmente para enviar un mensaje, solo es necesario seleccionar la conversación y escribir el mensaje.

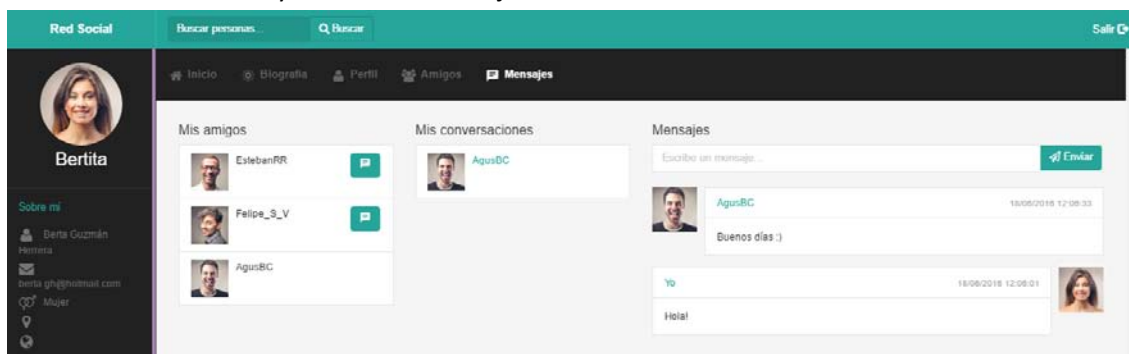
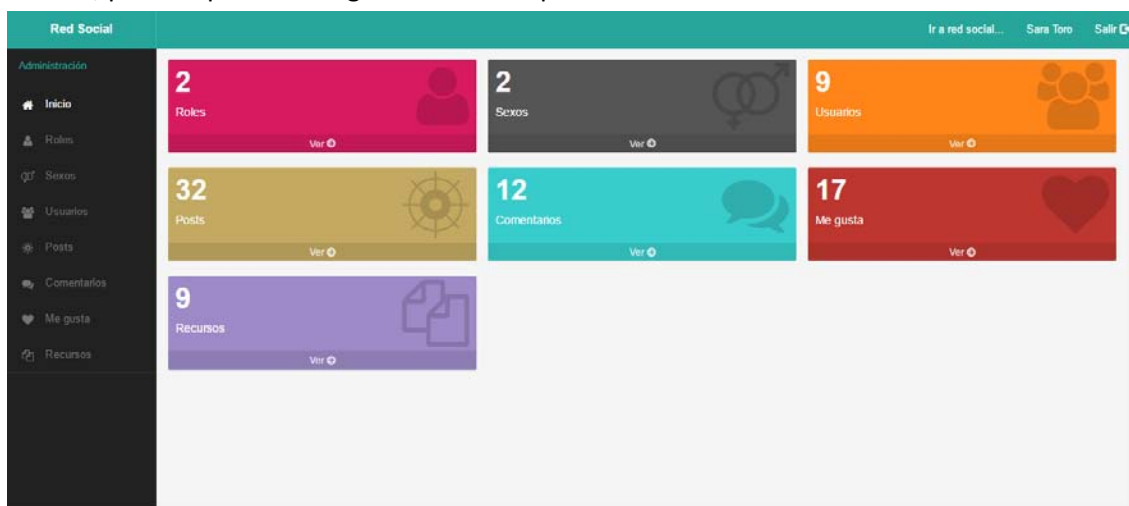


Ilustración 61: Interfaz mensajes

- **Administración**

- **Inicio**

Es la pantalla de resumen de la aplicación, en la que se muestran la cantidad de usuarios, comentarios, posts... que están registrados en la aplicación.



Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 62: Interfaz inicio - administración

La parte de administración está organizada de la misma manera siempre. Para cada grupo existen 4 opciones (ver todos, crear, editar y eliminar)

○ Roles

▪ Ver todos

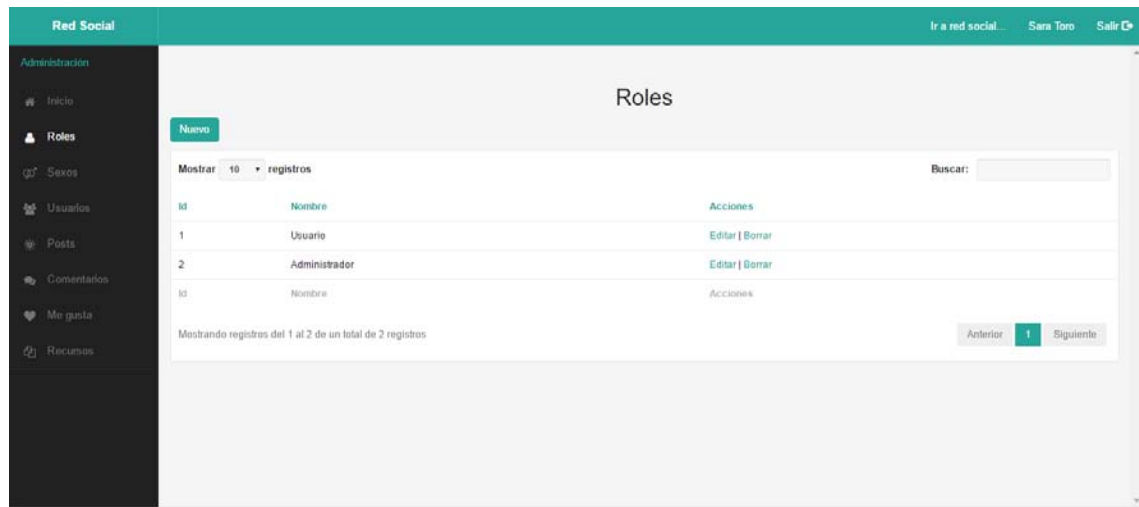


Ilustración 63: Interfaz ver todos los roles - administración

▪ Crear

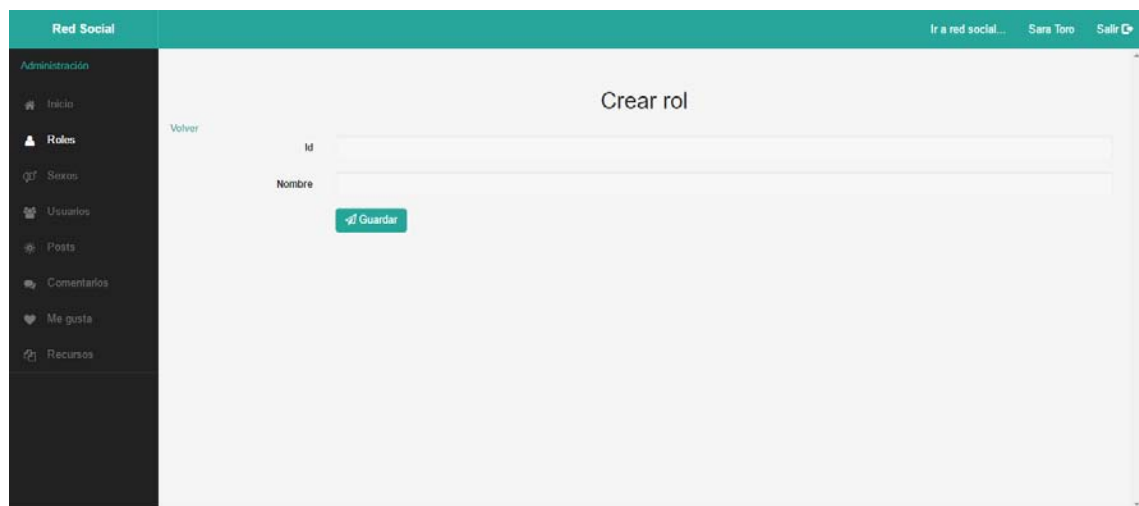


Ilustración 64: Interfaz crear rol - administración

■ Editar

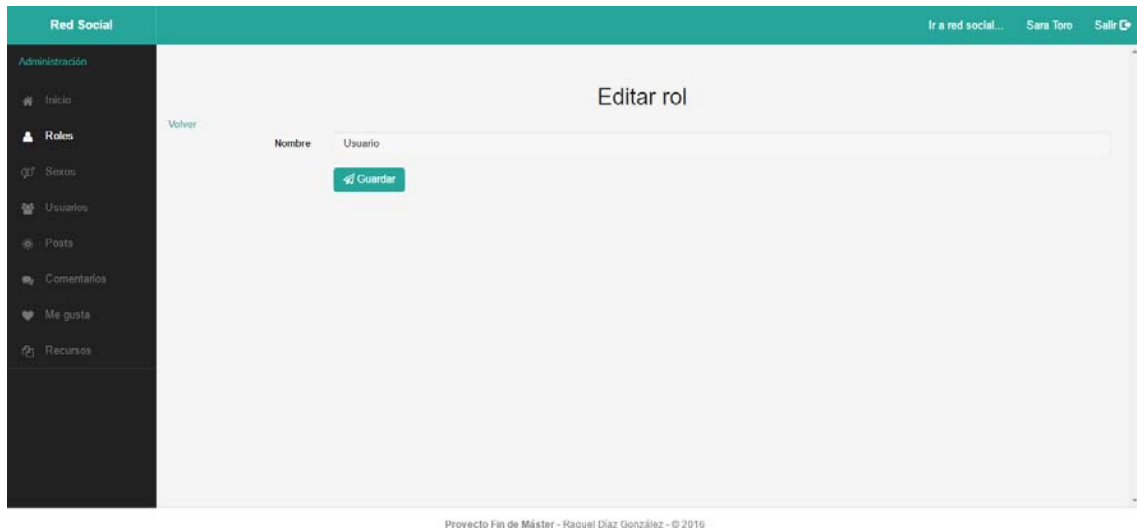


Ilustración 65: Interfaz editar rol - administración

■ Eliminar

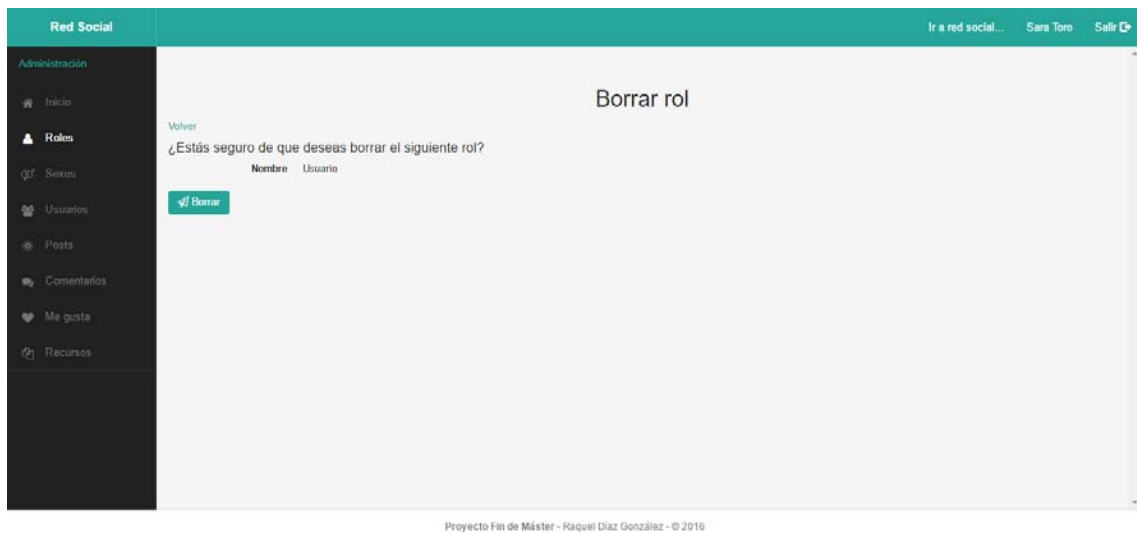
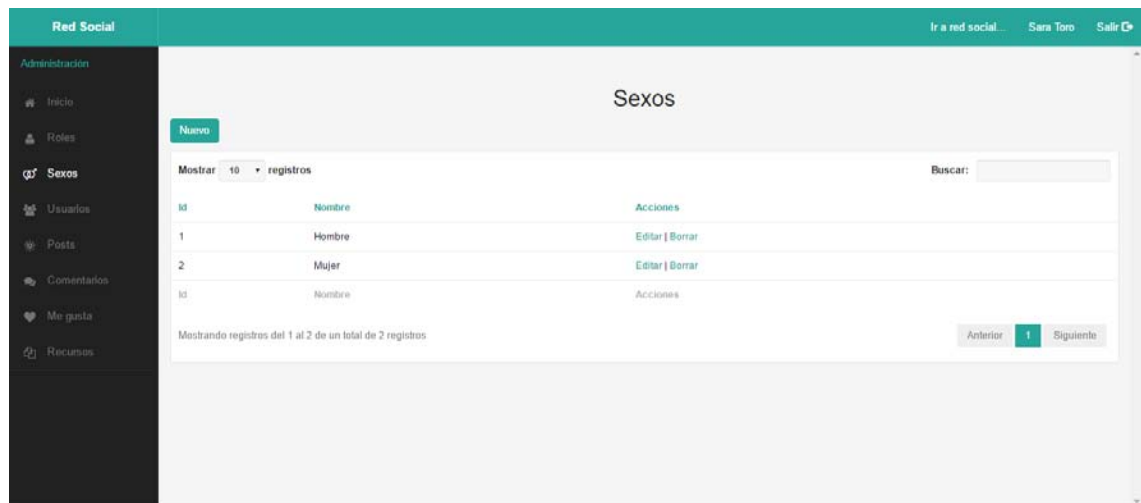


Ilustración 66: Interfaz eliminar rol - administración

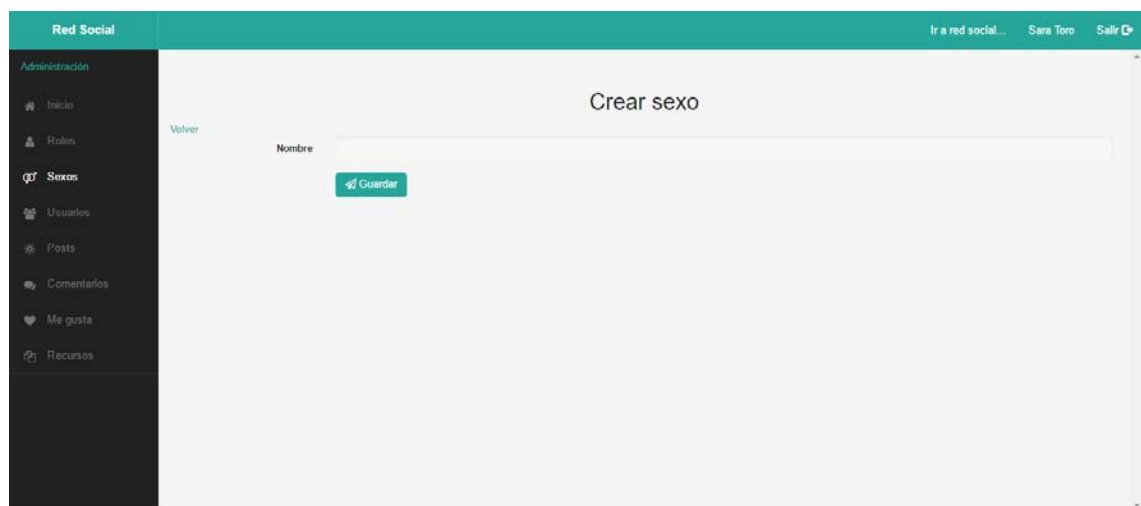
- Sexos
 - Ver todos



Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 67: Interfaz ver todos sexos - administración

- Crear



Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 68: Interfaz crear sexo - administración

■ Editar

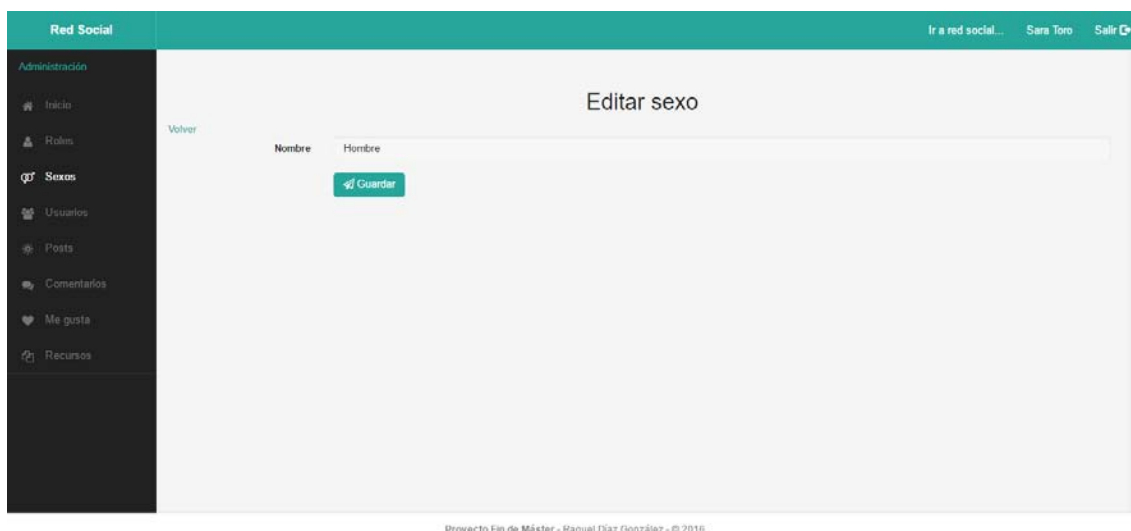


Ilustración 69: Interfaz editar sexo - administración

■ Eliminar



- **Usuarios**
 - **Ver todos**

Red Social										
<div> Ir a red social... Sara Toro Salir </div>										
Administración										
Inicio										
Roles										
Sexos										
Usuarios										
Posts										
Comentarios										
Me gusta										
Recursos										

Usuarios

Nuevo

Mostrar 10 registros

Buscar:

Id	Email	UserName	Livelo	Web	FullName	Sexo	Nº de posts	Nº de comentarios	Nº de me gusta	Acciones
18d1e41b-03bb-41ba-8829-990c26299859	alicia.cruz@gmail.com	AliCruz	Valencia		Alicia Cruz Flores	Mujer	3	1	2	Editar Borrar
31b3c9a9-9605-4c69-a8c6-45515b16637	sara.toro@gmail.com	Sara Toro	Madrid	www.saratororiz.com	Sara Toro Ortiz	Mujer	7	3	4	Editar Borrar
48c6534-6411-4d98-950-4a5e9b9d9c90	dani.mpino@gmail.com	Dani_Muñoz	Murcia		Daniel Muñoz Pino	Hombre	9	2	2	Editar Borrar
672a53a-d927-4f6b-806d-3e44fce0446	esteban.rr@gmail.com	EstebanRR			Esteban Rivera Ruiz	Hombre	2	0	0	Editar Borrar
70d2c46-a87e-4aa9-9c95-583035380cc0	felipe.sv@hotmail.com	Felipe_S_V	Pamplona		Felipe Soto Vergara	Hombre	4	1	3	Editar Borrar
7fee3117-9ecb-4262-b11b-6dac475cd124	c.martin.lopez@gmail.com	CristinaMartin			Cristina Martin Lopez	Mujer	1	0	2	Editar Borrar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 70: Interfaz ver todos usuarios - administración

- **Crear**

Red Social										
<div> Ir a red social... Sara Toro Salir </div>										
Administración										
Inicio										
Roles										
Sexos										
Usuarios										
Posts										
Comentarios										
Me gusta										
Recursos										

Crear usuario

Volver

Correo electrónico

Contraseña

Confirmar contraseña

Nombre de usuario

Nombre y apellidos

Sexo

Selecione...

AspNetRoles

Administrador
Usuario

Vivo en

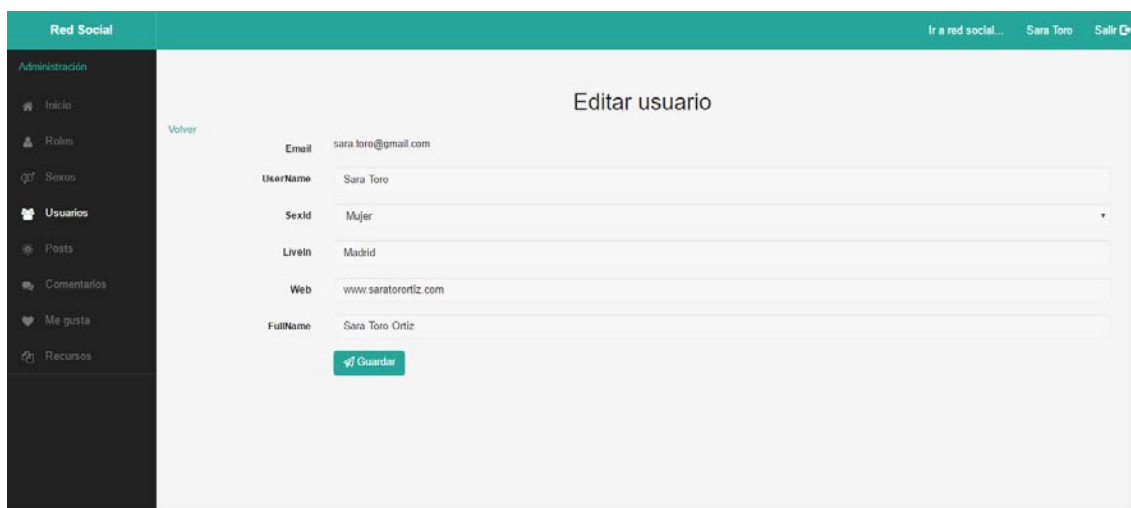
Web

Guardar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 71: Interfaz crear usuario - administración

■ Editar



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios**
- Posts
- Comentarios
- Me gusta
- Recursos

Editar usuario

Volver

Email sara.toro@gmail.com

UserName Sara Toro

SexId Mujer

LiveIn Madrid

Web www.saratorortiz.com

FullName Sara Toro Ortiz

Guardar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 72: Interfaz editar usuario - administración

■ Eliminar



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios**
- Posts
- Comentarios
- Me gusta
- Recursos

Borrar usuario

Volver

¿Estás seguro de que deseas borrar el siguiente usuario?

Email sara.toro@gmail.com

UserName Sara Toro

LiveIn Madrid

Web www.saratorortiz.com

FullName Sara Toro Ortiz

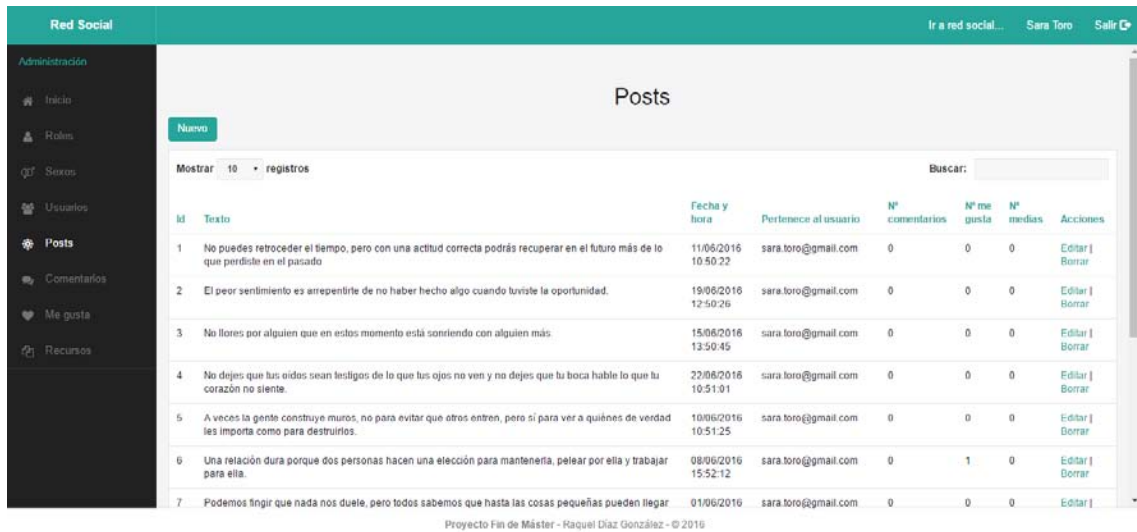
Nombre Mujer

Borrar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 73: Interfaz eliminar usuario - administración

- Posts
 - Ver todos



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios
- Posts**
- Comentarios
- Me gusta
- Recursos

Posts

Nuevo

Mostrar 10 registros

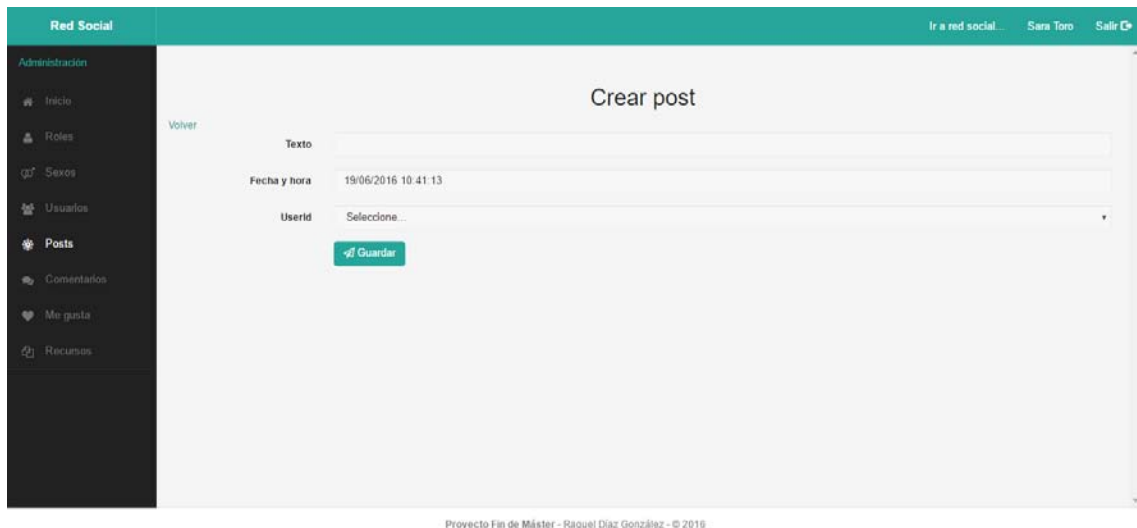
Buscar:

Id	Texto	Fecha y hora	Pertenece al usuario	Nº comentarios	Nº me gusta	Nº medias	Acciones
1	No puedes retroceder el tiempo, pero con una actitud correcta podrás recuperar en el futuro más de lo que perdiste en el pasado	11/06/2016 10:50:22	sara.toro@gmail.com	0	0	0	Editar Borrar
2	El peor sentimiento es arrepentirse de no haber hecho algo cuando tuviste la oportunidad.	19/06/2016 12:50:26	sara.toro@gmail.com	0	0	0	Editar Borrar
3	No llores por alguien que en estos momentos está sonriendo con alguien más.	15/06/2016 13:50:45	sara.toro@gmail.com	0	0	0	Editar Borrar
4	No dejes que tus oídos sean testigos de lo que tus ojos no ven y no dejes que tu boca hable lo que tu corazón no siente.	22/06/2016 10:51:01	sara.toro@gmail.com	0	0	0	Editar Borrar
5	A veces la gente construye muros, no para evitar que otros entren, pero sí para ver a quiénes de verdad les importa como para destruirlos.	10/06/2016 10:51:25	sara.toro@gmail.com	0	0	0	Editar Borrar
6	Una relación dura porque dos personas hacen una elección para mantenerla, pelear por ella y trabajar para ella.	08/06/2016 15:52:12	sara.toro@gmail.com	0	1	0	Editar Borrar
7	Podemos fingir que nada nos duele, pero todos sabemos que hasta las cosas pequeñas pueden llegar	01/06/2016	sara.toro@gmail.com	0	0	0	Editar Borrar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 74: Interfaz ver todos posts - administración

- Crear



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios
- Posts**
- Comentarios
- Me gusta
- Recursos

Crear post

Volver

Texto

Fecha y hora 19/06/2016 10:41:13

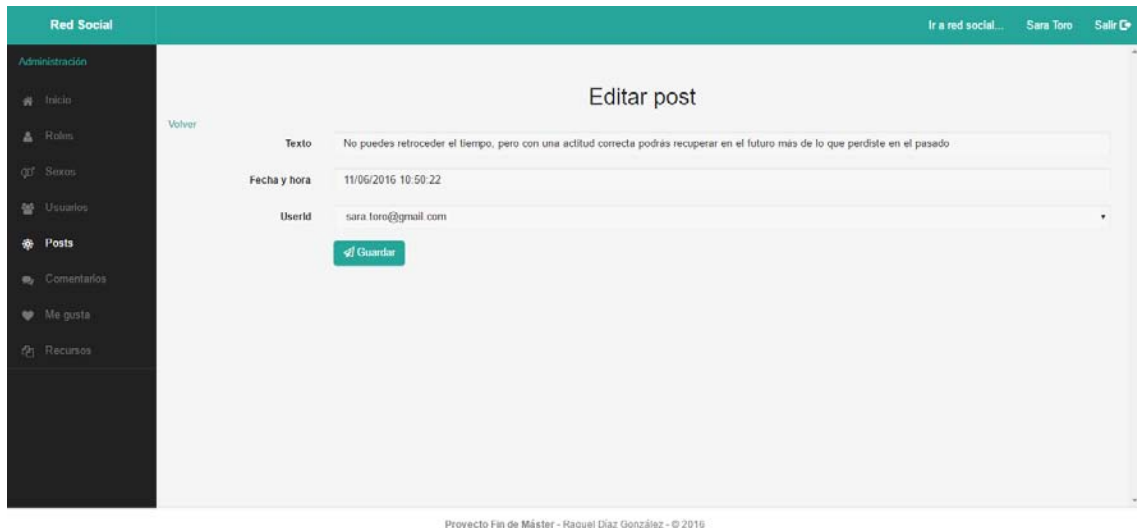
Usuario Selecciona...

Guardar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 75: Interfaz crear post - administración

■ Editar



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios
- Posts**
- Comentarios
- Me gusta
- Recursos

Editar post

Volver

Texto No puedes retroceder el tiempo, pero con una actitud correcta podrás recuperar en el futuro más de lo que perdiste en el pasado

Fecha y hora 11/06/2016 10:50:22

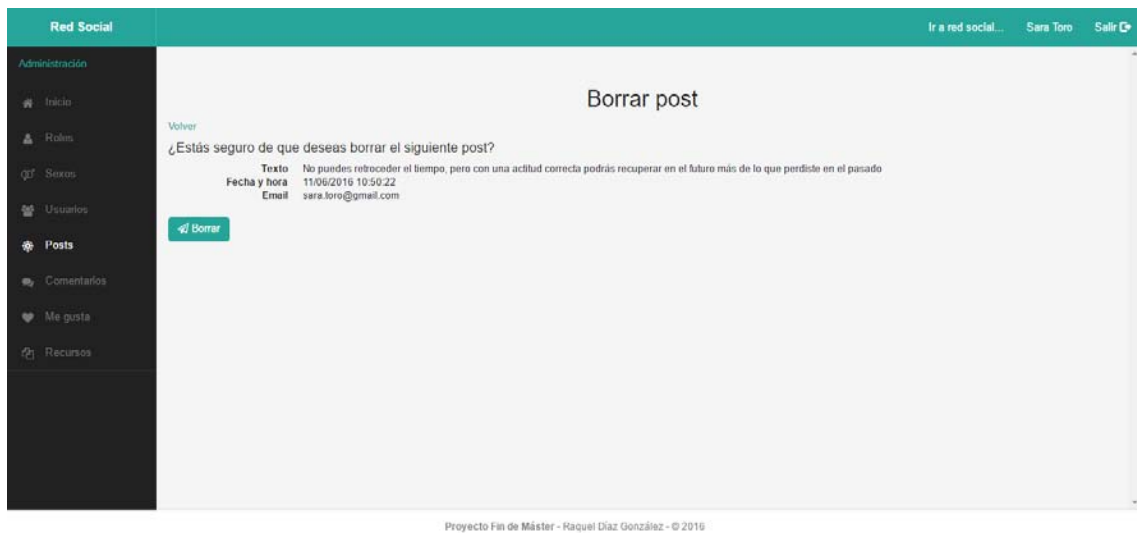
Userid sara.toro@gmail.com

Guardar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 76: Interfaz editar post - administración

■ Eliminar



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios
- Posts**
- Comentarios
- Me gusta
- Recursos

Borrar post

Volver

¿Estás seguro de que deseas borrar el siguiente post?

Texto No puedes retroceder el tiempo, pero con una actitud correcta podrás recuperar en el futuro más de lo que perdiste en el pasado

Fecha y hora 11/06/2016 10:50:22

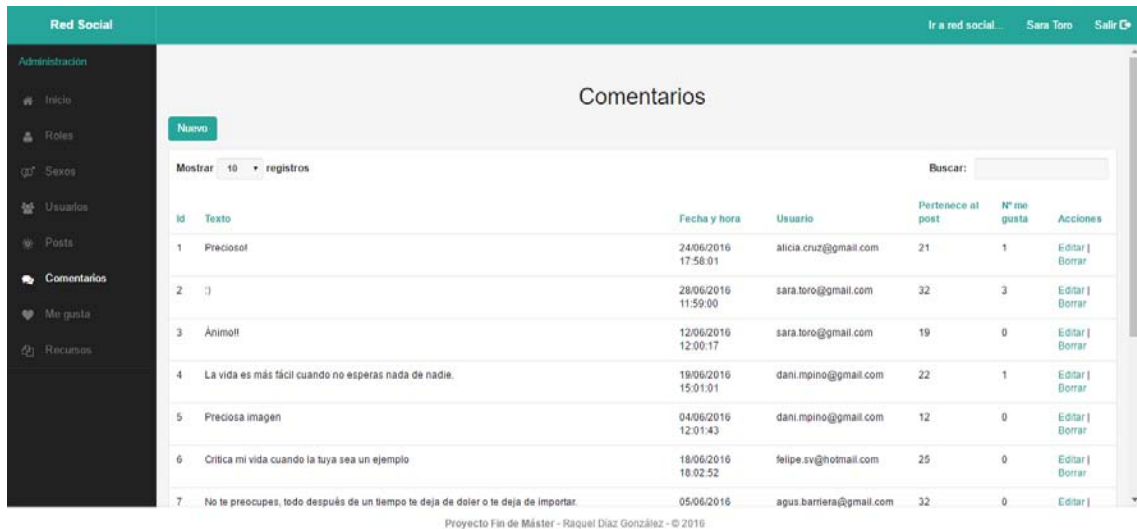
Email sara.toro@gmail.com

Borrar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 77: Interfaz borrar post - administración

- Comentarios
 - Ver todos

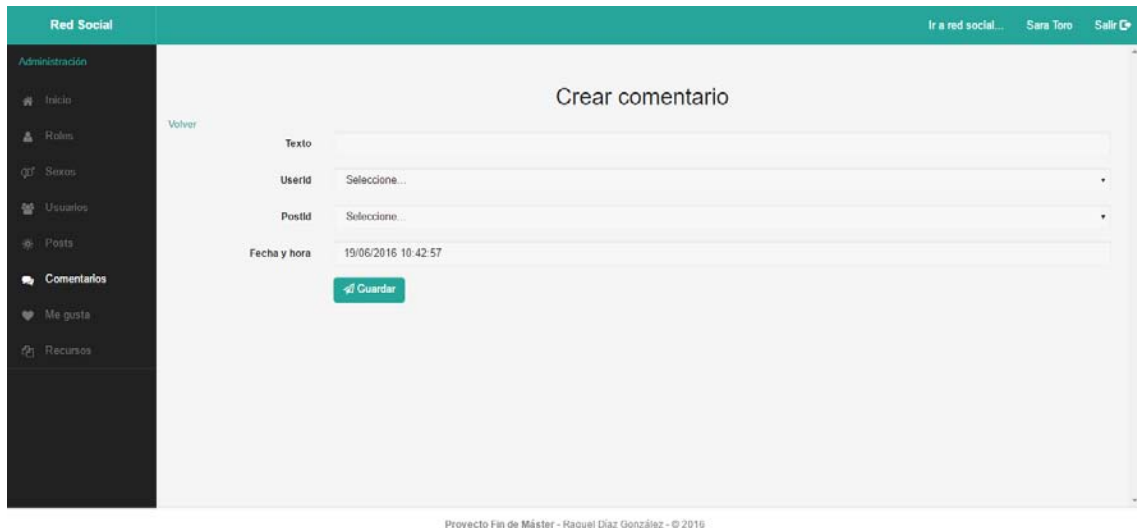


Id	Texto	Fecha y hora	Usuario	Pertenece al post	Nº me gusta	Acciones
1	Preciosos!	24/06/2016 17:58:01	alicia.cruz@gmail.com	21	1	Editar Borrar
2	:)	28/06/2016 11:59:00	sara.toro@gmail.com	32	3	Editar Borrar
3	Animott	12/06/2016 12:00:17	sara.toro@gmail.com	19	0	Editar Borrar
4	La vida es más fácil cuando no esperas nada de nadie.	19/06/2016 15:01:01	dani.mpino@gmail.com	22	1	Editar Borrar
5	Preciosa imagen	04/06/2016 12:01:43	dani.mpino@gmail.com	12	0	Editar Borrar
6	Crítica mi vida cuando la tuya sea un ejemplo	18/06/2016 18:02:52	felipe.sv@hotmail.com	25	0	Editar Borrar
7	No te preocupes, todo después de un tiempo te deja de doler o te deja de importar.	05/06/2016	agus.barrera@gmail.com	32	0	Editar Borrar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 78: Interfaz ver todos comentarios - administración

- Crear



Volver

Texto

Userid Seleccione...

Postid Seleccione...

Fecha y hora 19/06/2016 10:42:57

Guardar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 79: Interfaz crear comentario - administración

■ Editar



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios
- Posts
- Comentarios**
- Me gusta
- Recursos

Editar comentario

Volver

Texto: Precioso!

Usuario: alicia.cruz@gmail.com

Postid: Cuando dos personas están destinadas a estar juntas no importa donde estén o con quien estén, tarde o temprano se encontrarán.

Fecha y hora: 24/06/2016 17:58:01

[Guardar](#)

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 80: Interfaz editar comentario - administración

■ Eliminar



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios
- Posts
- Comentarios**
- Me gusta
- Recursos

Eliminar comentario

Volver

¿Estás seguro de que deseas borrar el siguiente comentario?

Texto	Precioso!
Fecha y hora	24/06/2016 17:58:01
Email	alicia.cruz@gmail.com

[Borrar](#)

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 81: Interfaz eliminar comentario - administración

- Me gusta
 - Ver todos

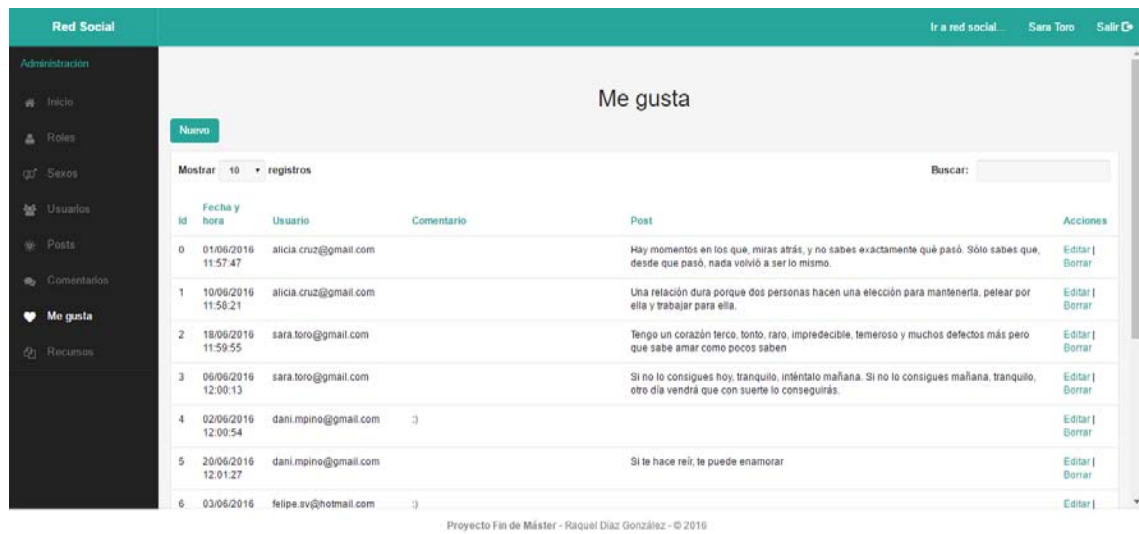


Ilustración 82: Interfaz ver todos me gusta - administración

- Crear

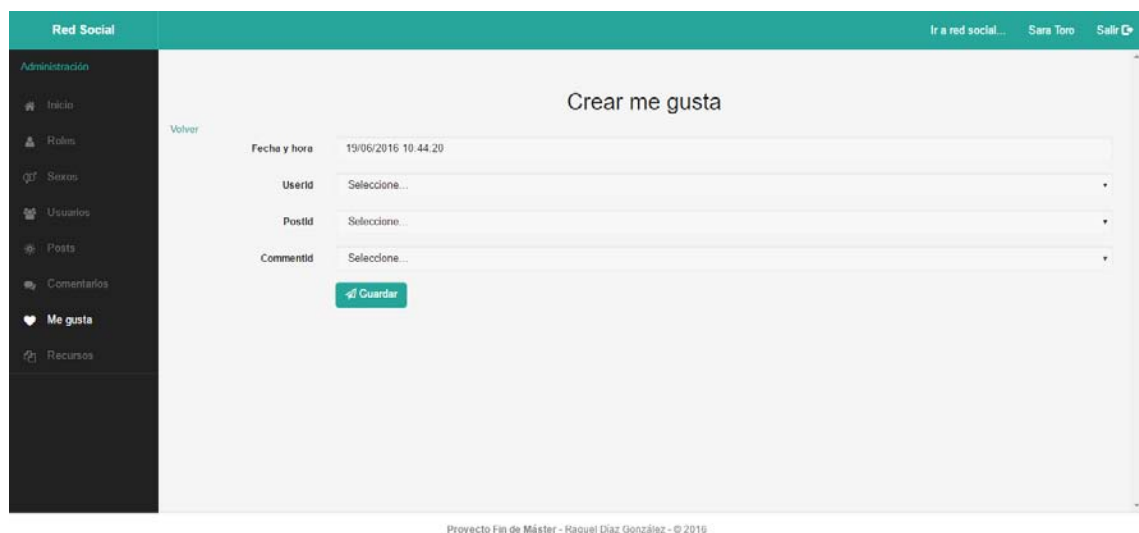


Ilustración 83: Interfaz crear me gusta - administración

■ Editar



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios
- Posts
- Comentarios
- Me gusta**
- Recursos

Editar me gusta

[Volver](#)

Fecha y hora: 01/06/2016 11:57:47

Usuario: alicia.cruz@gmail.com

Postid: Hay momentos en los que, miras atrás, y no sabes exactamente qué pasó. Sólo sabes que, desde que pasó, nada volvió a ser lo mismo.

Commentid: Selecciona...

[Guardar](#)

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 84: Interfaz editar me gusta - administración

■ Eliminar



Red Social Ir a red social... Sara Toro Salir

Administración

- Inicio
- Roles
- Sexos
- Usuarios
- Posts
- Comentarios
- Me gusta**
- Recursos

Borrar me gusta

[Volver](#)

¿Estás seguro de que deseas borrar el siguiente me gusta?

Fecha y hora: 01/06/2016 11:57:47

Email: alicia.cruz@gmail.com

Texto: Hay momentos en los que, miras atrás, y no sabes exactamente qué pasó. Sólo sabes que, desde que pasó, nada volvió a ser lo mismo.

[Borrar](#)

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 85: Interfaz eliminar me gusta - administración

- Recursos
 - Ver todos

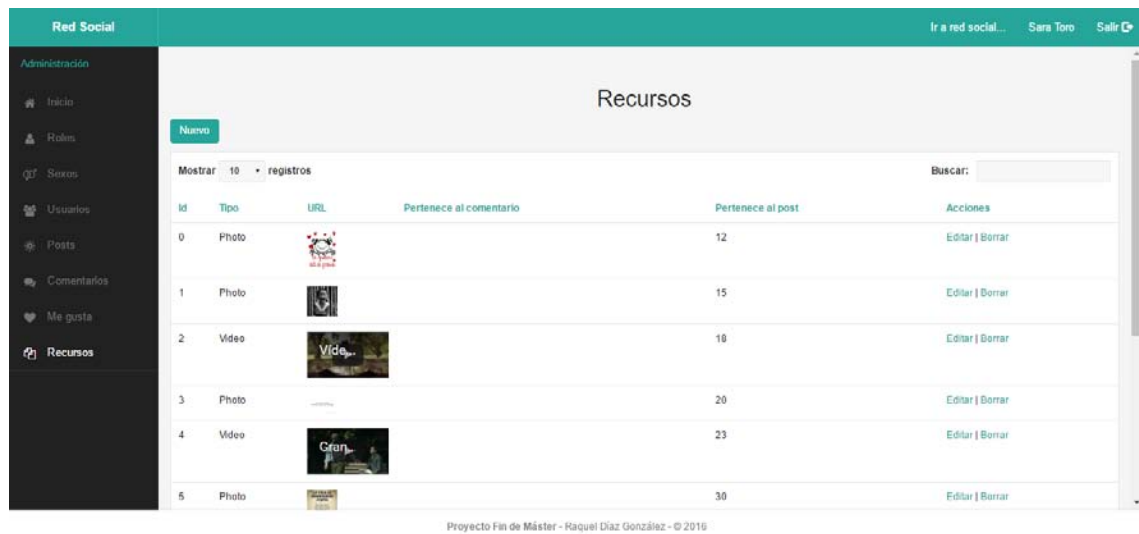


Ilustración 86: Interfaz ver todos recursos - administración

- Crear

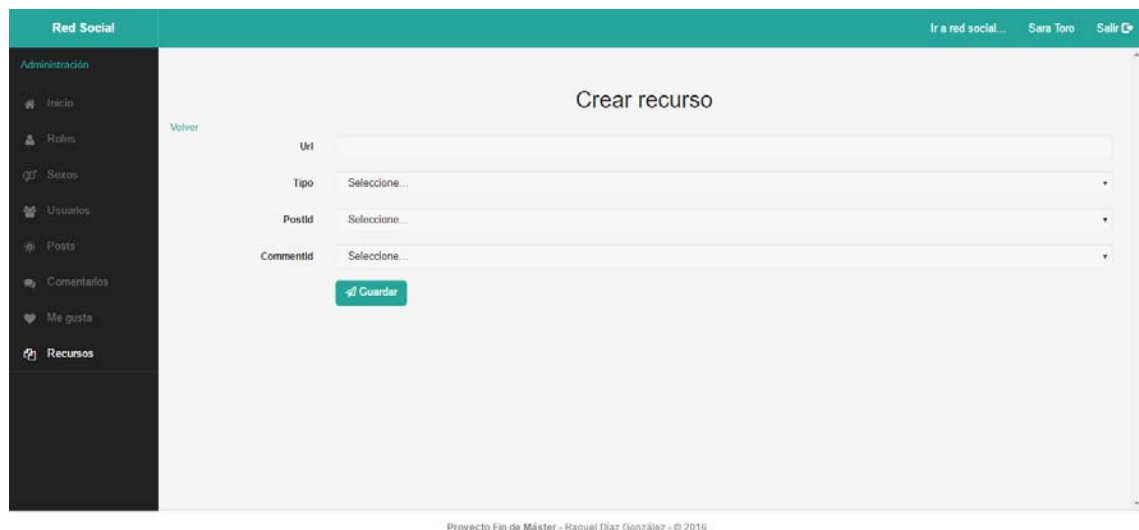



Ilustración 87: Interfaz crear recurso - administración

■ Editar



Red Social

Ir a red social... Sara Toro Salir

Administración

Inicio Roles Sexos Usuarios Posts Comentarios Me gusta Recursos

Volver

Editar recurso

Uri /Uploads/PostImages/20160618105513-tq.jpg

Tipo Photo

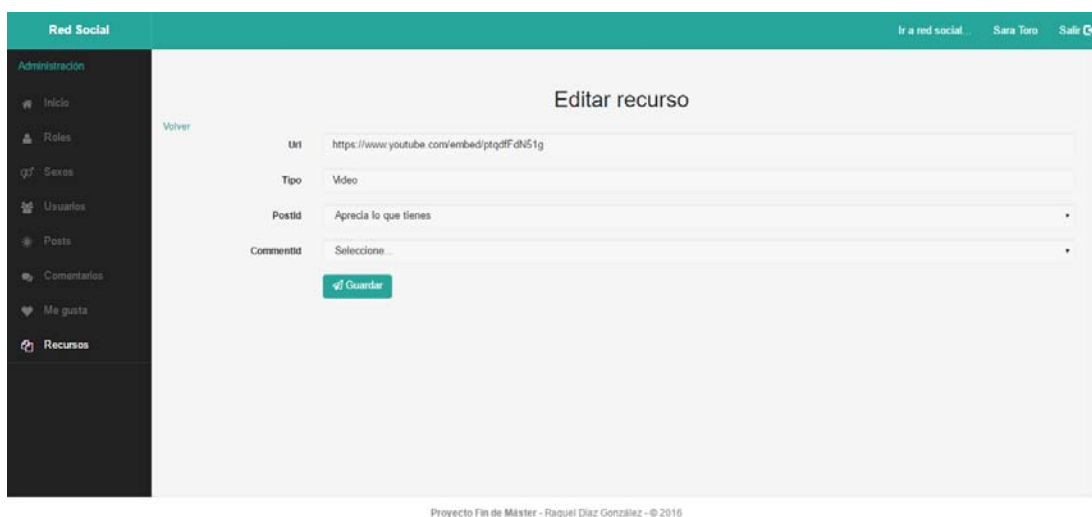
PostId Te quiero, sin importar las peleas. Te quiero, sin importar la distancia. Te quiero, sin importar los demás.

CommentId Seleccione...

Guardar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 88: Interfaz editar recurso (imagen) - administración



Red Social

Ir a red social... Sara Toro Salir

Administración

Inicio Roles Sexos Usuarios Posts Comentarios Me gusta Recursos

Volver

Editar recurso

Uri https://www.youtube.com/embed/ptgdfFdh51g

Tipo Video

PostId Aprecia lo que tienes.

CommentId Seleccione...

Guardar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 89: Interfaz editar recurso (vídeo) - administración

■ Eliminar



Red Social

Ir a red social... Sara Toro Salir

Administración

Inicio Roles Sexos Usuarios Posts Comentarios Me gusta Recursos

Volver

Borrar recurso

¿Estás seguro de que deseas borrar el siguiente recurso?

Uri /Uploads/PostImages/20160618105730-reir.jpg

Tipo Photo

Texto Si le hace reir, le puede enamorar

Borrar

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 90: Interfaz eliminar recurso - administración

Un punto muy importante que debe tenerse en cuenta en toda aplicación web es que la interfaz sea **responsive** para que pueda visualizarse correctamente desde cualquier dispositivo.

Debido a que se ha hecho uso de Bootstrap, este punto se cumple, y a continuación se pueden ver la misma interfaz en diferentes versiones responsive.

Laptop L (1440px)

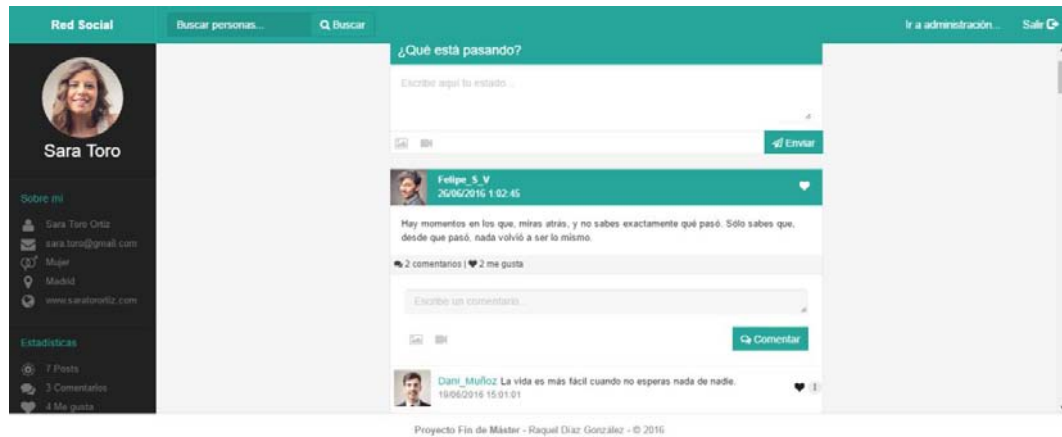


Ilustración 91: Interfaz laptop L

Laptop (1024px)

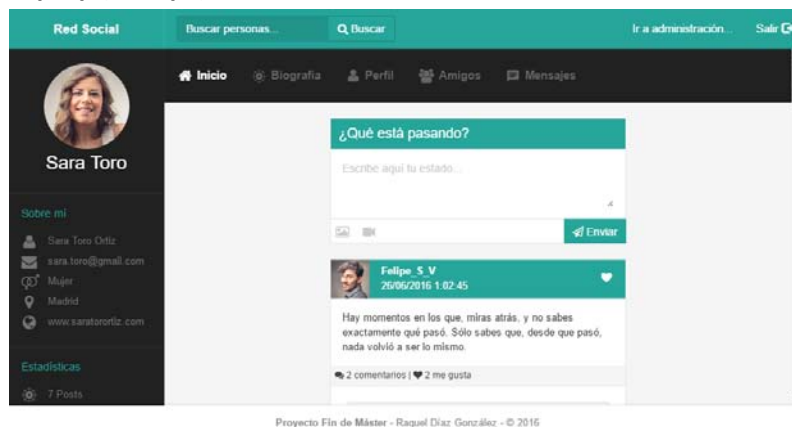


Ilustración 92: Interfaz laptop

Tablet (768px)

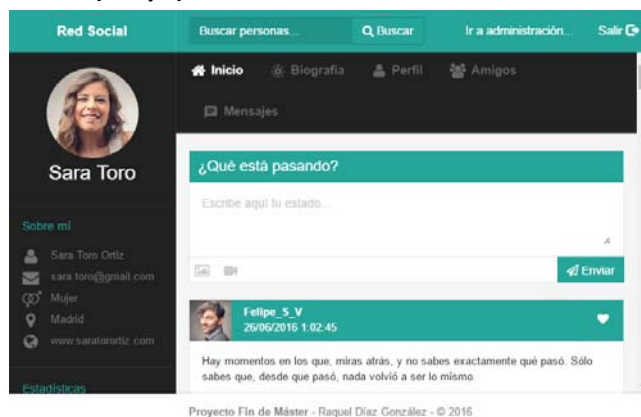
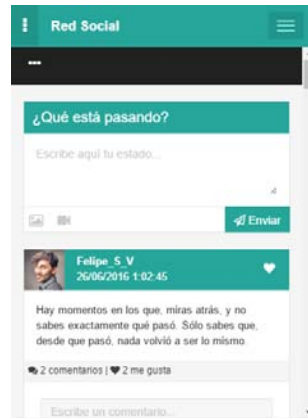


Ilustración 93: Interfaz Tablet

Mobile L (425px)

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Mobile M (375px)

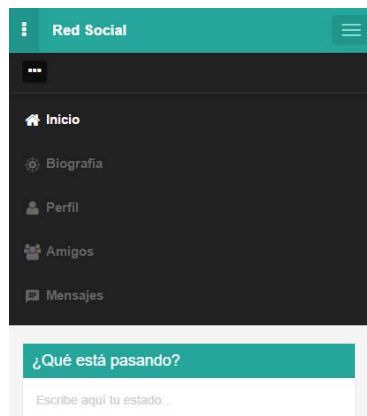
Proyecto Fin de Máster - Raquel Díaz González - © 2016

Mobile S (320px)

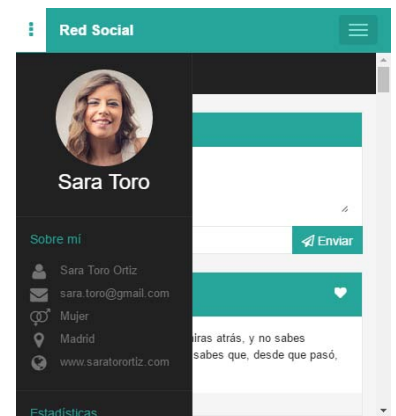
Proyecto Fin de Máster - Raquel Díaz González - © 2016

*Ilustración 94: Interfaces mobile L, M, S***Menú superior 1**

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Menú superior 2

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Menú lateral

Proyecto Fin de Máster - Raquel Díaz González - © 2016

Ilustración 95: Interfaces menú mobile



PRUEBAS

La **disciplina de pruebas** es el flujo de trabajo, incluyendo actividades, trabajadores y documentación, cuyo principal propósito es comprobar el resultado de la implementación al probar cada versión, incluyendo internas e intermedias, y versiones finales del sistema a entregar.

Clasificación de pruebas según el sujeto bajo prueba:

- Pruebas de Aceptación: usuarios, clientes e implicados (stakeholders) prueba la versión entregada, incluyendo requisitos funcionales y no funcionales
- Pruebas del Sistema: ingenieros de pruebas prueban las funciones del sistema como un todo
- Pruebas de Integración: son usadas para verificar que los componentes interactúan apropiadamente con otros después de que han sido integrados en una construcción
- Pruebas Unitarias: son para probar componentes como una unidad individual

CONCLUSIONES

Con el desarrollo de este trabajo de fin de máster, he conseguido alcanzar los objetivos propuestos al comienzo de la memoria.

Como se ha mencionado anteriormente, el desarrollo de este proyecto ha sido bastante importante tanto a nivel personal como profesional ya que me he enfrentado a un nuevo lenguaje que acababa de conocer en el máster y el resultado obtenido creo que ha sido fantástico.

La elección de la metodología ha sido la adecuada ya que la organización del proyecto por sprints ha ayudado notablemente a saber en cada momento las tareas que había que cumplir, y si se estaban cumpliendo en el tiempo estimado.

Por otro lado, las tecnologías utilizadas para el desarrollo del proyecto han aumentado bastante mis conocimientos y claramente ha sido un acierto elegir cada una de ellas.

Gracias a todo esto, el ritmo del desarrollo del proyecto ha sido bastante rápido porque no han existido días en los que hubiese grandes paradas y el hecho de no tener errores ayudaba que progresase adecuadamente.

Por último, he de decir que lo que más tiempo me llevó fue la creación del proyecto y la correspondiente organización en carpetas y subcarpetas.



POSIBLES AMPLIACIONES Y MEJORAS

Un software que está en uso nunca deja de tener ampliaciones y mejoras, por lo que si un día este proyecto comienza a ser utilizado, siempre se tendrá que poder dar soporte, solucionar fallos o incluir nuevas funcionalidades.

La falta de tiempo siempre es un factor que corre en nuestro contra, y por eso, me gustaría haber podido profundizar en más temas.

.NET permite el desarrollo de pruebas unitarias, así que habría sido un punto importante como para profundizar en él y poder así tener un conjunto de pruebas para cada caso de uso, ya que estamos ante un proyecto que es bastante grande.

Actualmente los posts, comentarios, me gusta, mensajes... se realizan de forma síncrona, lo que quiere decir que desde la vista se llama al controlador, y este controlador retorna la vista recargando la página. Esto en una red social no es lo más adecuado, pero por falta de tiempo y desconocimiento no se ha utilizado Ajax para realizar las peticiones que deben llamar a los controladores.

Actualmente tampoco existe ningún sistema de notificaciones, pero si este proyecto sigue adelante, es un punto clave que habría que implementar.

Los perfiles de los usuarios son todos públicos, así que lo adecuado sería dejar elegir al propio usuario si su perfil quiere que sea público o privado.

Entre otras funcionalidades futuras debería estar el recordar contraseña, sugerir amigos teniendo en cuenta los usuarios que se comparten entre dos amigos, realizar mejoras en la interfaz (como por ejemplo para publicar fotos y vídeos), implementar un chat, compartir ubicación, añadir emoticonos...



BIBLIOGRAFÍA

1. <https://msdn.microsoft.com>
2. <https://msdn.microsoft.com/es-es/library/>
3. <http://www.w3schools.com/aspnet/>
4. <http://stackoverflow.com/>
5. <http://anexsoft.com/p/46/como-subir-un-archivo-con-asp-net-mvc>
6. <http://www.codeproject.com/Articles/687061/Multiple-Models-in-a-View-in-ASP-NET-MVC-MVC>
7. <http://www.tutorialsteacher.com/mvc/mvc-folder-structure>
8. <http://www.dotnet-stuff.com/tutorials/aspnet-mvc/how-to-render-different-layout-in-asp-net-mvc>
9. https://datatables.net/examples/advanced_init/language_file.html

